

[illegible]

```
SSSSSSSS EEEEEEEEE TTTT TTTT
SSSSSSSS EEEEEEEEE TTTT TTTT
SS      EE      TT
SS      EE      TT
SS      EE      TT
SS      EE      TT
SSSSSS   EEEEEEEE TT
SSSSSS   EEEEEEEE TT
          EE      TT
          EE      TT
          EE      TT
          EE      TT
SSSSSSSS EEEEEEEEE TT
SSSSSSSS EEEEEEEEE TT
          EEEEEEEE TT
          EEEEEEEE TT
```

```
LL      IIIIII SSSSSSSS
LL      IIIIII SSSSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSSS
LL      II     SSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLL IIIIII SSSSSSSS
```

```
0001 0 MODULE MAIL$SET( XTITLE 'VAX/VMS MAIL UTILITY SET COMMAND'
0002 0 IDENT = 'V04-000' ) =
0003 1 BEGIN
0004 1
0005 1
0006 1 *****
0007 1 *
0008 1 *   COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 *   DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 *   ALL RIGHTS RESERVED.
0011 1 *
0012 1 *   THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 *   ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 *   INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 *   COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 *   OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 *   TRANSFERRED.
0018 1 *
0019 1 *   THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 *   AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 *   CORPORATION.
0022 1 *
0023 1 *   DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 *   SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *****
0027 1
0028 1
0029 1
0030 1 ++
0031 1 FACILITY:      VAX/VMS MAIL UTILITY
0032 1
0033 1 ABSTRACT:      SET command handling.
0034 1
0035 1 AUTHOR:        Benn Schreiber, 5-Dec-1982
0036 1
0037 1 MODIFIED BY:
0038 1
0039 1     V03-010 BLS0255      Benn Schreiber      28-Dec-1983
0040 1             New global flags
0041 1
0042 1     V03-009 BLS0246      Benn Schreiber      8-Nov-1983
0043 1             Rework for new VMSMAIL.DAT handling, move new mail
0044 1             count to VMSMAIL.
0045 1
0046 1     V03-008 BLS0229      Benn Schreiber      29-Jul-1983
0047 1             Add SET NOMAIL_DIRECTORY command
0048 1
0049 1     V03-007 BLS0225      Benn Schreiber      17-Jun-1983
0050 1             Minimize window of confusion in SET MAIL_DIRECTORY. Enforce
0051 1             restriction that personal name must start with alphabetic
0052 1             character (informal personal name spec agreement with
0053 1             RSX and ARPA-interface mail developers).
0054 1
0055 1     V03-006 BLS0222      Benn Schreiber      7-May-1983
0056 1             Disallow setting the wastebasket folder to MAIL or NEWMAIL
0057 1
```

MAIL\$SET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

M 8
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]SET.B32;1

Page 2
(1)

..	58	0058	1	!	V03-005	BLS0220	Benn Schreiber	30-Apr-1983
..	59	0059	1	!		Add SET MAIL_DIRECTORY,SET AUTO_PURGE		
..	60	0060	1	!				
..	61	0061	1	!	V03-004	BLS0217	Benn Schreiber	12-Apr-1983
..	62	0062	1	!		Add SET COPY_SELF		
..	63	0063	1	!				
..	64	0064	1	!	V03-003	BLS0211	Benn Schreiber	13-Mar-1983
..	65	0065	1	!		New commands for ISAM mail files		
..	66	0066	1	!				
..	67	0067	1	!	V03-002	BLS0202	Benn Schreiber	5-Jan-1982
..	68	0068	1	!		Check strings for illegal length/characters		
..	69	0069	1	!--				


```
71 0070 1 |
72 0071 1 | Declarations
73 0072 1 |
74 0073 1 | LIBRARY 'SYSS$LIBRARY:LIB';
75 0074 1 | REQUIRE 'SRC$:MAILREQ';
76 0220 1 | LIBRARY 'LIB$:MAILDEF';
77 0221 1 |
78 0222 1 | OWN
79 0223 1 |     COMMAND_DESC: DYN_DESC_DECL,
80 0224 1 |     DYN_DESC: DYN_DESC_DECL,
81 0225 1 |     DYN_DESC_2: DYN_DESC_DECL;
82 0226 1 |
83 0227 1 | EXTERNAL
84 0228 1 |     MAIL$RD_RAB : $BBLOCK,           |MAIL file RAB
85 0229 1 |     MAIL$GC_FLAGS : $BBLOCK,         |Control flags
86 0230 1 |     MAIL$SD_MAIL : $BBLOCK,          |Descriptor of 'MAIL'
87 0231 1 |     MAIL$SD_NEWMAIL : $BBLOCK,       |and 'NEWMAIL'
88 0232 1 |     MAIL$SD_SETF1: $BBLOCK,          |Descriptor for set cmd prompt
89 0233 1 |     MAIL$SD_SETN1: $BBLOCK,
90 0234 1 |     MAIL$SD_USER: $BBLOCK,           |Descriptor for USER
91 0235 1 |     MAIL$Q_RD_DESC : $BBLOCK,        |Current mail file descriptor
92 0236 1 |     MAIL$T_UAF_BUFF : REF $BBLOCK,   |UAF buffer
93 0237 1 |     MAIL$Q_USERDEF : $BBLOCK,        |dev/dir for mail files
94 0238 1 |     MAIL$Q_MAILDEF : $BBLOCK,        |above + 'mail.mai'
95 0239 1 |     MAIL$Q_WASTEDESC : $BBLOCK,      |Descriptor for waste folder
96 0240 1 |     MAIL$Q_USERNAME: $BBLOCK,        |Descriptor for username
97 0241 1 |     MAIL$Q_FULLUSER: $BBLOCK,        |and for full name
98 0242 1 |     MAIL$T_SYSMAIRAB: $BBLOCK,       |RAB for VMSMAIL.DAT
99 0243 1 |     MAIL$T_MYUIC,                    |UIC of person running MAIL
100 0244 1 |     MAIL$T_USRSYSBUF: REF $BBLOCK,   |User's vmsmail record
101 0245 1 |     MAIL$T_SYSMABUF: REF $BBLOCK;    |Pointer to buffer
102 0246 1 |
103 0247 1 | EXTERNAL ROUTINE
104 0248 1 |     CLI$GET_VALUE: ADDRESSING_MODE(GENERAL), |Get value from CLI
105 0249 1 |     CLI$PRESENT: ADDRESSING_MODE(GENERAL), |Test qualifier present
106 0250 1 |     LIB$CREATE_DIR,                   |Create a directory
107 0251 1 |     LIB$RENAME_FILE,                 |Rename files
108 0252 1 |     MAIL$GETLK_SYSMAL_REC,           |Read and lock VMSMAIL record
109 0253 1 |     MAIL$MOVE_WASTEBASKET,           |Move contents of wastebasket
110 0254 1 |     MAIL$OPEN_IF,                    |Open file if not open
111 0255 1 |     MAIL$UPCASE_Q,                   |Cvt to upper case, honor quotes
112 0256 1 |     MAIL$UPDATE_INFO_REC,            |Update mail file info rec
113 0257 1 |     MAIL$UPDATE_SYSMAL_REC,          |Update VMSMAIL record
114 0258 1 |     STR$APPEND,                       |Append a string
115 0259 1 |     STR$CONCAT: ADDRESSING_MODE(GENERAL), |Concatenate strings
116 0260 1 |     STR$COPY_DX: ADDRESSING_MODE(GENERAL), |Copy string
117 0261 1 |     STR$FREE_DX: ADDRESSING_MODE(GENERAL), |Free dynamic string
118 0262 1 |     MAIL$GET_USER,                   |Get user uaf record
119 0263 1 |     MAIL$OPEN_FILE,                  |Open a mail file
120 0264 1 |     MAIL$FORW_PRIV_CHECK,            |See if SYSNAM privilege
121 0265 1 |     UTIL$GETFILENAME,                |Get file spec from FAB
122 0266 1 |     UTIL$REPORT_IO_ERROR;            |Report I/O error
123 0267 1 |
124 0268 1 | | Declare names for string descriptors
125 0269 1 |
126 0270 1 | GLOBAL BIND
127 0271 1 |     MAIL$Q_MAIL_MAI = $DESCRIPTOR('MAIL.MAI') : $BBLOCK;
```

```
128      0272 1
129 P 0273 1 SD(
130 P 0274 1      'FILE',
131 P 0275 1      'LOG',
132 P 0276 1      'NOSEND',
133 P 0277 1      'NOREPLY',
134 P 0278 1      'SEND',
135 P 0279 1      'REPLY',
136      0280 1      'WASTENAME');
137      0281 1
138      0282 1      ! Define shared messages
139      0283 1
140 P 0284 1 $SHR_MSGDEF(MAIL,126,LOCAL,
141 P 0285 1      (DIRNOTCRE,ERROR),
142 P 0286 1      (EXISTS,INFO),
143 P 0287 1      (RENAMED,INFO),
144 P 0288 1      (CREATED,INFO),
145 P 0289 1      (PARSEFAIL,ERROR),
146 P 0290 1      (SEARCHFAIL,ERROR),
147      0291 1      (WRITEERR,WARNING));
148      0292 1 LITERAL
149      0293 1      PRO_SYS      = $BITPOSITION(XAB$V_SYS),
150      0294 1      PRO_OWN      = $BITPOSITION(XAB$V_OWN),
151      0295 1      PRO_GRP      = $BITPOSITION(XAB$V_GRP),
152      0296 1      PRO_WLD      = $BITPOSITION(XAB$V_WLD),
153      0297 1
154      0298 1      DIR_PRO      =
155      0299 1      (
156      0300 1      (
157      0301 1      (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_SYS OR
158      0302 1      (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_OWN OR
159      0303 1      (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_GRP OR
160      0304 1      (XAB$M_NOREAD OR XAB$M_NOWRITE OR XAB$M_NOEXE OR XAB$M_NODEL) ^ PRO_WLD;
161      0305 1      )
162      0306 1      )
163      0307 1      )
164      0308 1      )
165      0309 1      )
166      0310 1      )
167      0311 1      )
168      0312 1      )
169      0313 1      )
170      0314 1      )
171      0315 1      )
172      0316 1      )
173      0317 1      )
174      0318 1      )
175      0319 1      )
176      0320 1      )
177      0321 1      )
178      0322 1      )
179      0323 1      )
180      0324 1      )
181      0325 1      )
182      0326 1      )
183      0327 1      )
184      0328 1      )
185      0329 1      )
186      0330 1      )
187      0331 1      )
188      0332 1      )
189      0333 1      )
190      0334 1      )
191      0335 1      )
192      0336 1      )
193      0337 1      )
194      0338 1      )
195      0339 1      )
196      0340 1      )
197      0341 1      )
198      0342 1      )
199      0343 1      )
200      0344 1      )
201      0345 1      )
202      0346 1      )
203      0347 1      )
204      0348 1      )
205      0349 1      )
206      0350 1      )
207      0351 1      )
208      0352 1      )
209      0353 1      )
210      0354 1      )
211      0355 1      )
212      0356 1      )
213      0357 1      )
214      0358 1      )
215      0359 1      )
216      0360 1      )
217      0361 1      )
218      0362 1      )
219      0363 1      )
220      0364 1      )
221      0365 1      )
222      0366 1      )
223      0367 1      )
224      0368 1      )
225      0369 1      )
226      0370 1      )
227      0371 1      )
228      0372 1      )
229      0373 1      )
230      0374 1      )
231      0375 1      )
232      0376 1      )
233      0377 1      )
234      0378 1      )
235      0379 1      )
236      0380 1      )
237      0381 1      )
238      0382 1      )
239      0383 1      )
240      0384 1      )
241      0385 1      )
242      0386 1      )
243      0387 1      )
244      0388 1      )
245      0389 1      )
246      0390 1      )
247      0391 1      )
248      0392 1      )
249      0393 1      )
250      0394 1      )
251      0395 1      )
252      0396 1      )
253      0397 1      )
254      0398 1      )
255      0399 1      )
256      0400 1      )
257      0401 1      )
258      0402 1      )
259      0403 1      )
260      0404 1      )
261      0405 1      )
262      0406 1      )
263      0407 1      )
264      0408 1      )
265      0409 1      )
266      0410 1      )
267      0411 1      )
268      0412 1      )
269      0413 1      )
270      0414 1      )
271      0415 1      )
272      0416 1      )
273      0417 1      )
274      0418 1      )
275      0419 1      )
276      0420 1      )
277      0421 1      )
278      0422 1      )
279      0423 1      )
280      0424 1      )
281      0425 1      )
282      0426 1      )
283      0427 1      )
284      0428 1      )
285      0429 1      )
286      0430 1      )
287      0431 1      )
288      0432 1      )
289      0433 1      )
290      0434 1      )
291      0435 1      )
292      0436 1      )
293      0437 1      )
294      0438 1      )
295      0439 1      )
296      0440 1      )
297      0441 1      )
298      0442 1      )
299      0443 1      )
300      0444 1      )
301      0445 1      )
302      0446 1      )
303      0447 1      )
304      0448 1      )
305      0449 1      )
306      0450 1      )
307      0451 1      )
308      0452 1      )
309      0453 1      )
310      0454 1      )
311      0455 1      )
312      0456 1      )
313      0457 1      )
314      0458 1      )
315      0459 1      )
316      0460 1      )
317      0461 1      )
318      0462 1      )
319      0463 1      )
320      0464 1      )
321      0465 1      )
322      0466 1      )
323      0467 1      )
324      0468 1      )
325      0469 1      )
326      0470 1      )
327      0471 1      )
328      0472 1      )
329      0473 1      )
330      0474 1      )
331      0475 1      )
332      0476 1      )
333      0477 1      )
334      0478 1      )
335      0479 1      )
336      0480 1      )
337      0481 1      )
338      0482 1      )
339      0483 1      )
340      0484 1      )
341      0485 1      )
342      0486 1      )
343      0487 1      )
344      0488 1      )
345      0489 1      )
346      0490 1      )
347      0491 1      )
348      0492 1      )
349      0493 1      )
350      0494 1      )
351      0495 1      )
352      0496 1      )
353      0497 1      )
354      0498 1      )
355      0499 1      )
356      0500 1      )
357      0501 1      )
358      0502 1      )
359      0503 1      )
360      0504 1      )
361      0505 1      )
362      0506 1      )
363      0507 1      )
364      0508 1      )
365      0509 1      )
366      0510 1      )
367      0511 1      )
368      0512 1      )
369      0513 1      )
370      0514 1      )
371      0515 1      )
372      0516 1      )
373      0517 1      )
374      0518 1      )
375      0519 1      )
376      0520 1      )
377      0521 1      )
378      0522 1      )
379      0523 1      )
380      0524 1      )
381      0525 1      )
382      0526 1      )
383      0527 1      )
384      0528 1      )
385      0529 1      )
386      0530 1      )
387      0531 1      )
388      0532 1      )
389      0533 1      )
390      0534 1      )
391      0535 1      )
392      0536 1      )
393      0537 1      )
394      0538 1      )
395      0539 1      )
396      0540 1      )
397      0541 1      )
398      0542 1      )
399      0543 1      )
400      0544 1      )
401      0545 1      )
402      0546 1      )
403      0547 1      )
404      0548 1      )
405      0549 1      )
406      0550 1      )
407      0551 1      )
408      0552 1      )
409      0553 1      )
410      0554 1      )
411      0555 1      )
412      0556 1      )
413      0557 1      )
414      0558 1      )
415      0559 1      )
416      0560 1      )
417      0561 1      )
418      0562 1      )
419      0563 1      )
420      0564 1      )
421      0565 1      )
422      0566 1      )
423      0567 1      )
424      0568 1      )
425      0569 1      )
426      0570 1      )
427      0571 1      )
428      0572 1      )
429      0573 1      )
430      0574 1      )
431      0575 1      )
432      0576 1      )
433      0577 1      )
434      0578 1      )
435      0579 1      )
436      0580 1      )
437      0581 1      )
438      0582 1      )
439      0583 1      )
440      0584 1      )
441      0585 1      )
442      0586 1      )
443      0587 1      )
444      0588 1      )
445      0589 1      )
446      0590 1      )
447      0591 1      )
448      0592 1      )
449      0593 1      )
450      0594 1      )
451      0595 1      )
452      0596 1      )
453      0597 1      )
454      0598 1      )
455      0599 1      )
456      0600 1      )
457      0601 1      )
458      0602 1      )
459      0603 1      )
460      0604 1      )
461      0605 1      )
462      0606 1      )
463      0607 1      )
464      0608 1      )
465      0609 1      )
466      0610 1      )
467      0611 1      )
468      0612 1      )
469      0613 1      )
470      0614 1      )
471      0615 1      )
472      0616 1      )
473      0617 1      )
474      0618 1      )
475      0619 1      )
476      0620 1      )
477      0621 1      )
478      0622 1      )
479      0623 1      )
480      0624 1      )
481      0625 1      )
482      0626 1      )
483      0627 1      )
484      0628 1      )
485      0629 1      )
486      0630 1      )
487      0631 1      )
488      0632 1      )
489      0633 1      )
490      0634 1      )
491      0635 1      )
492      0636 1      )
493      0637 1      )
494      0638 1      )
495      0639 1      )
496      0640 1      )
497      0641 1      )
498      0642 1      )
499      0643 1      )
500      0644 1      )
501      0645 1      )
502      0646 1      )
503      0647 1      )
504      0648 1      )
505      0649 1      )
506      0650 1      )
507      0651 1      )
508      0652 1      )
509      0653 1      )
510      0654 1      )
511      0655 1      )
512      0656 1      )
513      0657 1      )
514      0658 1      )
515      0659 1      )
516      0660 1      )
517      0661 1      )
518      0662 1      )
519      0663 1      )
520      0664 1      )
521      0665 1      )
522      0666 1      )
523      0667 1      )
524      0668 1      )
525      0669 1      )
526      0670 1      )
527      0671 1      )
528      0672 1      )
529      0673 1      )
530      0674 1      )
531      0675 1      )
532      0676 1      )
533      0677 1      )
534      0678 1      )
535      0679 1      )
536      0680 1      )
537      0681 1      )
538      0682 1      )
539      0683 1      )
540      0684 1      )
541      0685 1      )
542      0686 1      )
543      0687 1      )
544      0688 1      )
545      0689 1      )
546      0690 1      )
547      0691 1      )
548      0692 1      )
549      0693 1      )
550      0694 1      )
551      0695 1      )
552      0696 1      )
553      0697 1      )
554      0698 1      )
555      0699 1      )
556      0700 1      )
557      0701 1      )
558      0702 1      )
559      0703 1      )
560      0704 1      )
561      0705 1      )
562      0706 1      )
563      0707 1      )
564      0708 1      )
565      0709 1      )
566      0710 1      )
567      0711 1      )
568      0712 1      )
569      0713 1      )
570      0714 1      )
571      0715 1      )
572      0716 1      )
573      0717 1      )
574      0718 1      )
575      0719 1      )
576      0720 1      )
577      0721 1      )
578      0722 1      )
579      0723 1      )
580      0724 1      )
581      0725 1      )
582      0726 1      )
583      0727 1      )
584      0728 1      )
585      0729 1      )
586      0730 1      )
587      0731 1      )
588      0732 1      )
589      0733 1      )
590      0734 1      )
591      0735 1      )
592      0736 1      )
593      0737 1      )
594      0738 1      )
595      0739 1      )
596      0740 1      )
597      0741 1      )
598      0742 1      )
599      0743 1      )
600      0744 1      )
601      0745 1      )
602      0746 1      )
603      0747 1      )
604      0748 1      )
605      0749 1      )
606      0750 1      )
607      0751 1      )
608      0752 1      )
609      0753 1      )
610      0754 1      )
611      0755 1      )
612      0756 1      )
613      0757 1      )
614      0758 1      )
615      0759 1      )
616      0760 1      )
617      0761 1      )
618      0762 1      )
619      0763 1      )
620      0764 1      )
621      0765 1      )
622      0766 1      )
623      0767 1      )
624      0768 1      )
625      0769 1      )
626      0770 1      )
627      0771 1      )
628      0772 1      )
629      0773 1      )
630      0774 1      )
631      0775 1      )
632      0776 1      )
633      0777 1      )
634      0778 1      )
635      0779 1      )
636      0780 1      )
637      0781 1      )
638      0782 1      )
639      0783 1      )
640      0784 1      )
641      0785 1      )
642      0786 1      )
643      0787 1      )
644      0788 1      )
645      0789 1      )
646      0790 1      )
647      0791 1      )
648      0792 1      )
649      0793 1      )
650      0794 1      )
651      0795 1      )
652      0796 1      )
653      0797 1      )
654      0798 1      )
655      0799 1      )
656      0800 1      )
657      0801 1      )
658      0802 1      )
659      0803 1      )
660      0804 1      )
661      0805 1      )
662      0806 1      )
663      0807 1      )
664      0808 1      )
665      0809 1      )
666      0810 1      )
667      0811 1      )
668      0812 1      )
669      0813 1      )
670      0814 1      )
671      0815 1      )
672      0816 1      )
673      0817 1      )
674      0818 1      )
675      0819 1      )
676      0820 1      )
677      0821 1      )
678      0822 1      )
679      0823 1      )
680      0824 1      )
681      0825 1      )
682      0826 1      )
683      0827 1      )
684      0828 1      )
685      0829 1      )
686      0830 1      )
687      0831 1      )
688      0832 1      )
689      0833 1      )
690      0834 1      )
691      0835 1      )
692      0836 1      )
693      0837 1      )
694      0838 1      )
695      0839 1      )
696      0840 1      )
697      0841 1      )
698      0842 1      )
699      0843 1      )
700      0844 1      )
701      0845 1      )
702      0846 1      )
703      0847 1      )
704      0848 1      )
705      0849 1      )
706      0850 1      )
707      0851 1      )
708      0852 1      )
709      0853 1      )
710      0854 1      )
711      0855 1      )
712      0856 1      )
713      0857 1      )
714      0858 1      )
715      0859 1      )
716      0860 1      )
717      0861 1      )
718      0862 1      )
719      0863 1      )
720      0864 1      )
721      0865 1      )
722      0866 1      )
723      0867 1      )
724      0868 1      )
725      0869 1      )
726      0870 1      )
727      0871 1      )
728      0872 1      )
729      0873 1      )
730      0874 1      )
731      0875 1      )
732      0876 1      )
733      0877 1      )
734      0878 1      )
735      0879 1      )
736      0880 1      )
737      0881 1      )
738      0882 1      )
739      0883 1      )
740      0884 1      )
741      0885 1      )
742      0886 1      )
743      0887 1      )
744      0888 1      )
745      0889 1      )
746      0890 1      )
747      0891 1      )
748      0892 1      )
749      0893 1      )
750      0894 1      )
751      0895 1      )
752      0896 1      )
753      0897 1      )
754      0898 1      )
755      0899 1      )
756      0900 1      )
757      0901 1      )
758      0902 1      )
759      0903 1      )
760      0904 1      )
761      0905 1      )
762      0906 1      )
763      0907 1      )
764      0908 1      )
765      0909 1      )
766      0910 1      )
767      0911 1      )
768      0912 1      )
769      0913 1      )
770      0914 1      )
771      0915 1      )
772      0916 1      )
773      0917 1      )
774      0918 1      )
775      0919 1      )
776      0920 1      )
777      0921 1      )
778      0922 1      )
779      0923 1      )
780      0924 1      )
781      0925 1      )
782      0926 1      )
783      0927 1      )
784      0928 1      )
785      0929 1      )
786      0930 1      )
787      0931 1      )
788      0932 1      )
789      0933 1      )
790      0934 1      )
791      0935 1      )
792      0936 1      )
793      0937 1      )
794      0938 1      )
795      0939 1      )
796      0940 1      )
797      0941 1      )
798      0942 1      )
799      0943 1      )
800      0944 1      )
801      0945 1      )
802      0946 1      )
803      0947 1      )
804      0948 1      )
805      0949 1      )
806      0950 1      )
807      0951 1      )
808      0952 1      )
809      0953 1      )
810      0954 1      )
811      0955 1      )
812      0956 1      )
813      0957 1      )
814      0958 1      )
815      0959 1      )
816      0960 1      )
817      0961 1      )
818      0962 1      )
819      0963 1      )
820      0964 1      )
821      0965 1      )
822      0966 1      )
823      0967 1      )
824      0968 1      )
825      0969 1      )
826      0970 1      )
827      0971 1      )
828      0972 1      )
829      0973 1      )
830      0974 1      )
831      0975 1      )
832      0976 1      )
833      0977 1      )
834      0978 1      )
835      0979 1      )
836      0980 1      )
837      0981 1      )
838      0982 1      )
839      0983 1      )
840      0984 1      )
841      0985 1      )
842      0986 1      )
843      0987 1      )
844      0988 1      )
845      0989 1      )
846      0990 1      )
847      0991 1      )
848      0992 1      )
849      0993 1      )
850      0994 1      )
851      0995 1      )
852      0996 1      )
853      0997 1      )
854      0998 1      )
855      0999 1      )
856      1000 1      )
857      1001 1      )
858      1002 1      )
859      1003 1      )
860      1004 1      )
861      1005 1      )
862      1006 1      )
863      1007 1      )
864      1008 1      )
865      1009 1      )
866      1010 1      )
867      1011 1      )
868      1012 1      )
869      1013 1      )
870      1014 1      )
871      1015 1      )
872      1016 1      )
873      1017 1      )
874      1018 1      )
875      1019 1      )
876      1020 1      )
877      1021 1      )
878      1022 1      )
879      1023 1      )
880      1024 1      )
881      1025 1      )
882      1026 1      )
883      1027 1      )
884      1028 1      )
885      1029 1      )
886      1030 1      )
887      1031 1      )
888      1032 1      )
889      1033 1      )
890      1034 1      )
891      1035 1      )
892      1036 1      )
893      1037 1      )
894      1038 1      )
895      1039 1      )
896      1040 1      )
897      1041 1      )
898      1042 1      )
899      1043 1      )
900      1044 1      )
901      1045 1      )
902      1046 1      )
903      1047 1      )
904      1048 1      )
905      1049 1      )
906      1050 1      )
907      1051 1      )
908      1052 1      )
909      1053 1      )
910      1054 1      )
911      1055 1      )
912      1056 1      )
913      1057 1      )
914      1058 1      )
915      1059 1      )
916      1060 1      )
917      1061 1      )
918      1062 1      )
919      1063 1      )
920      1064 1      )
921      1065 1      )
922      1066 1      )
923      1067 1      )
924      1068 1      )
925      1069 1      )
926      1070 1      )
927      1071 1      )
928      1072 1      )
929      1073 1      )
930      1074 1      )
931      1075 1      )
932      1076 1      )
933      1077 1      )
934      1078 1      )
935      1079 1      )
936      1080 1      )
937      1081 1      )
938      1082 1      )
939      1083 1      )
940      1084 1      )
941      1085 1      )
942      1086 1      )
943      1087 1      )
944      1088 1      )
945      1089 1      )
946      1090 1      )
947      1091 1      )
948      1092 1      )
949      1093 1      )
950      1094 1      )
951      1095 1      )
952      1096 1      )
953      1097 1      )
954      1098 1      )
955      1099 1      )
956      1100 1      )
957      1101 1      )
958      1102 1      )
959      1103 1      )
960      1104 1      )
961      1105 1      )
962      1106 1      )
963      1107 1      )
964      1108 1      )
965      1109 1      )
966      1110 1      )
967      1111 1      )
968      1112 1      )
969      1113 1      )
970      1114 1      )
971      1115 1      )
972      1116 1      )
973      1117 1      )
974      1118 
```



```
160 0303 1 GLOBAL ROUTINE MAIL$STRIP_QUOTES(OUT_DESC,IN_DESC) =
161 0304 1 ++
162 0305 1 FUNCTIONAL DESCRIPTION:
163 0306 1
164 0307 1 This routine strips double quotes (if any) from the input string
165 0308 1
166 0309 1 INPUTS:
167 0310 1
168 0311 1 IN_DESC = address of string descriptor for input string
169 0312 1 OUT_DESC = address of string descriptor for output string
170 0313 1
171 0314 1 --
172 0315 2 BEGIN
173 0316 2 MAP
174 0317 2 IN_DESC : REF $BBLOCK,
175 0318 2 OUT_DESC : REF $BBLOCK;
176 0319 2
177 0320 2 LOCAL
178 0321 2 DESC_1 : $BBLOCK[DSC$C_S_BLN],
179 0322 2 CHAR : BYTE;
180 0323 2
181 0324 2 LITERAL
182 0325 2 DOUBLE_QUOTE = %ASCII '"';
183 0326 2
184 0327 2 DESC_1[DSC$W_LENGTH] = .IN_DESC[DSC$W_LENGTH];
185 0328 2 DESC_1[DSC$B_CLASS] = 0;
186 0329 2 DESC_1[DSC$B_DTYPE] = 0;
187 0330 2 DESC_1[DSC$A_POINTER] = .IN_DESC[DSC$A_POINTER];
188 0331 2
189 0332 2 IF .DESC_1[DSC$W_LENGTH] GTRU 0
190 0333 3 THEN BEGIN
191 0334 3 CHAR = CH$RCHAR(.DESC_1[DSC$A_POINTER]);
192 0335 3 IF .DESC_1[DSC$W_LENGTH] GTR 1
193 0336 4 AND (.CHAR EQL DOUBLE_QUOTE)
194 0337 4 AND (.CHAR EQL CH$RCHAR(.DESC_1[DSC$A_POINTER]+
195 0338 4 .DESC_1[DSC$W_LENGTH] - 1))
196 0339 4 THEN BEGIN
197 0340 4 DESC_1[DSC$W_LENGTH] = .DESC_1[DSC$W_LENGTH] - 2;
198 0341 4 DESC_1[DSC$A_POINTER] = .DESC_1[DSC$A_POINTER] + 1;
199 0342 3 END;
200 0343 2 END;
201 0344 2
202 0345 2 RETURN STR$COPY_DX(.OUT_DESC,DESC_1)
203 0346 1 END;
```

```
.TITLE MAIL$SET VAX/VMS MAIL UTILITY SET COMMAND
.IDENT \V04-000\
```

```
.PSECT $CODE$,NOWRT,2
```

```
49 41 4D 2E 4C 49 41 4D 00000 P.AAB: .ASCII \MAIL.MAI\
00000008 00008 P.AAA: .LONG 8
00000000 0000C .ADDRESS P.AAB
45 4C 49 46 00010 P.AAD: .ASCII \FILE\
00000004 00014 P.AAC: .LONG 4
00000000 00018 .ADDRESS P.AAD
```

```

      47 4F 4C 0001C P.AAF: .ASCII \LOG\
      0001F .BLKB 1
      00000003 00020 P.AAE: .LONG 3
      00000000 00024 .ADDRESS P.AAF
      44 4E 45 53 4F 4E 00028 P.AAH: .ASCII \NOSEND\
      0002E .BLKB 2
      00000006 00030 P.AAG: .LONG 6
      00000000 00034 .ADDRESS P.AAH
      59 4C 50 45 52 4F 4E 00038 P.AAJ: .ASCII \NOREPLY\
      0003F .BLKB 1
      00000007 00040 P.AAI: .LONG 7
      00000000 00044 .ADDRESS P.AAJ
      44 4E 45 53 00048 P.AAL: .ASCII \SEND\
      00000004 0004C P.AAK: .LONG 4
      00000000 00050 .ADDRESS P.AAL
      59 4C 50 45 52 00054 P.AAN: .ASCII \REPLY\
      00059 .BLKB 3
      00000005 0005C P.AAM: .LONG 5
      00000000 00060 .ADDRESS P.AAN
      45 4D 41 4E 45 54 53 41 57 00064 P.AAP: .ASCII \WASTENAME\
      0006D .BLKB 3
      00000009 00070 P.AAO: .LONG 9
      00000000 00074 .ADDRESS P.AAP
```

.PSECT \$OWNS,NOEXE,2

```

00# 00000 COMMAND_DESC:
      02 0E 00002 .BYTE 0[2]
      00004 .BYTE 14, 2
      00008 DYN_DESC:
      02 0E 0000A .BYTE 0[2]
      0000C .BYTE 14, 2
      00010 DYN_DESC_2:
      02 0E 00012 .BYTE 0[2]
      00014 .BYTE 14, 2
      .BLKB 4
```

```

MAIL$Q MAIL_MAI== P.AAA
SD_FILE= P.AAC
SD_LOG= P.AAE
SD_NOSEND= P.AAG
SD_NOREPLY= P.AAI
SD_SEND= P.AAK
SD_REPLY= P.AAM
SD_WASTENAME= P.AAO
      .EXTRN MAIL$RD_RAB, MAIL$GL_FLAGS
      .EXTRN MAIL$SD_MAIL, MAIL$SD_NEWMAIL
      .EXTRN MAIL$SD_SETF1, MAIL$SD_SETN1
      .EXTRN MAIL$SD_USER, MAIL$Q_RD_DESC
      .EXTRN MAIL$T_OAF_BUFF
      .EXTRN MAIL$Q_USERDEF, MAIL$Q_MAILDEF
      .EXTRN MAIL$Q_WASTEDESC
      .EXTRN MAIL$Q_USERNAME
      .EXTRN MAIL$Q_FULLUSER
      .EXTRN MAIL$T_SYSMAIRAB
```


					.EXTRN	MAIL\$MYUIC, MAIL\$USRSYBUF	
					.EXTRN	MAIL\$SYSMAIBUF	
					.EXTRN	CLISGET VALUE, CLISPRESENT	
					.EXTRN	LIB\$CREATE_DIR, LIB\$RENAME_FILE	
					.EXTRN	MAIL\$GETLK-SYSMAI_REC	
					.EXTRN	MAIL\$MOVE_WASTEBASKET	
					.EXTRN	MAIL\$OPEN_IF, MAIL\$UPCASE_Q	
					.EXTRN	MAIL\$UPDATE_INFO_REC	
					.EXTRN	MAIL\$UPDATE-SYSMAI_REC	
					.EXTRN	STR\$APPEND, STR\$CONCAT	
					.EXTRN	STR\$COPY_DX, STR\$FREE1_DX	
					.EXTRN	MAIL\$GET_USER, MAIL\$OPEN_FILE	
					.EXTRN	MAIL\$FORM_PRIV_CHECK	
					.EXTRN	UTIL\$GETFILENAME	
					.EXTRN	UTIL\$REPORT_IO_ERROR	
					.PSECT	\$CODE\$,NOWRT,2	
					.ENTRY	MAIL\$STRIP_QUOTES, Save nothing	0303
					SUBL2	#4, SP	
					MOVL	IN_DESC, R0	0327
					MOVZWL	(R0), DESC_1	
					MOVL	4(R0), DESC_1+4	0330
					MOVZWL	DESC_1, R0	0332
					BEQL	1\$	
					MOVB	@DESC_1+4, CHAR	0334
					CMPW	R0, #T	0335
					BLEQU	1\$	
					CMPB	CHAR, #34	0336
					BNEQ	1\$	
					ADDL2	DESC_1+4, R0	0338
					CMPB	CHAR, -1(R0)	
					BNEQ	1\$	
					SUBW2	#2, DESC_1	0340
					INCL	DESC_1+4-	0341
					PUSHL	SP	0345
					PUSHL	OUT_DESC	
					CALLS	#2, STR\$COPY_DX	
					RET		0346

				0000 00000	
				04 C2 00002	
				AC D0 00005	
				60 3C 00009	
04				A0 D0 0000C	
				6E 3C 00011	
				1E 13 00014	
				BE 90 00016	
				50 B1 0001A	
				15 1B 0001D	
				51 91 0001F	
				10 12 00022	
				AE C0 00024	
FF				51 91 00028	
				06 12 0002C	
				02 A2 0002E	
				AE D6 00031	
				5E DD 00034 1\$:	
				AC DD 00036	
				02 FB 00039	
				04 00040	
00000000G	00				

; Routine Size: 65 bytes, Routine Base: \$CODE\$ + 0078

```
205 0347 1 ROUTINE CHECK_STRING(DESC,MAXLENGTH) =
206 0348 1 ++
207 0349 1 FUNCTIONAL DESCRIPTION:
208 0350 1
209 0351 1 Check a string for too long or illegal characters
210 0352 1
211 0353 1 INPUTS:
212 0354 1
213 0355 1 DESC = Addr of string descriptor for string
214 0356 1 MAXLENGTH = Max length allowable for string
215 0357 1
216 0358 1 ROUTINE VALUE:
217 0359 1
218 0360 1 true if all ok, else error
219 0361 1 --
220 0362 2 BEGIN
221 0363 2
222 0364 2 MAP
223 0365 2 DESC : REF $BLOCK;
224 0366 2
225 0367 2 LOCAL
226 0368 2 CHAR : BYTE,
227 0369 2 PTR;
228 0370 2
229 0371 2
230 0372 2 Check length of string
231 0373 2
232 0374 2 IF .DESC[DSC$W_LENGTH] GTRU .MAXLENGTH
233 0375 2 THEN BEGIN
234 0376 2 SIGNAL(MAIL$_NAMTOOBIG,1,.DESC);
235 0377 2 RETURN MAIL$_NAMTOOBIG
236 0378 2 END;
237 0379 2
238 0380 2 Check for characters with high bit on or illegal character (less than
239 0381 2 a space, and not a tab)
240 0382 2
241 0383 2 PTR = .DESC[DSC$A_POINTER];
242 0384 2 INCRU I FROM 1 TO .DESC[DSC$W_LENGTH]
243 0385 2 DO IF ((CHAR = CHRCHAR_A(PTR) AND ZX'80') NEQ 0
244 0386 2 OR ((CHAR LSSU XC' ') AND (.CHAR NEQ XC' ')))
245 0387 2 THEN BEGIN
246 0388 2 SIGNAL(MAIL$_ILLCHAR,3,.CHAR,.DESC[DSC$W_LENGTH],.DESC[DSC$A_POINTER]);
247 0389 2 RETURN MAIL$_ILLCHAR
248 0390 2 END;
249 0391 2
250 0392 2 RETURN TRUE
251 0393 1 END;
```

```
00FC 00000 CHECK_STRING:
57 00000000G 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7
52 04 AC D0 00009 MOVAB LIB$SIGNAL, R7
10 00 ED 0000D MOVL DESC, R2
OB AC 62 CMPZV #0, #16, (R2), MAXLENGTH
```

```
: 0347
: 0374
:
```

		15	1B	00013	BLEQU	1\$	
		52	DD	00015	PUSHL	R2	0376
		01	DD	00017	PUSHL	#1	
	007E8092	8F	DD	00019	PUSHL	#8290450	
67		03	FB	0001F	CALLS	#3, LIB\$SIGNAL	
50	007E8092	8F	DD	00022	MOVL	#8290450, R0	0377
			04	00029	RET		
56	04	A2	DD	0002A	MOVL	4(R2), PTR	0383
54		62	3C	0002E	MOVZWL	(R2), R4	0384
55		01	DD	00031	MOVL	#1, }	
		2F	11	00034	BRB	5\$	
50		86	9A	00036	MOVZBL	(PTR)+, R0	0385
53		50	90	00039	MOVB	R0, CHAR	
		0A	19	0003C	BLSS	3\$	
20		53	91	0003E	CMPB	CHAR, #32	0386
		20	1E	00041	BGEQU	4\$	
09		53	91	00043	CMPB	CHAR, #9	
		1B	13	00046	BEQL	4\$	
	04	A2	DD	00048	PUSHL	4(R2)	0388
		54	DD	0004B	PUSHL	R4	
7E		53	9A	0004D	MOVZBL	CHAR, -(SP)	
		03	DD	00050	PUSHL	#3	
	007E809A	8F	DD	00052	PUSHL	#8290458	
67		05	FB	00058	CALLS	#5, LIB\$SIGNAL	
50	007E809A	8F	DD	0005B	MOVL	#8290458, R0	0389
			04	00062	RET		
		55	D6	00063	INCL	1	0385
54		55	D1	00065	CMPL	1, R4	
		CC	1B	00068	BLEQU	2\$	
50		01	DD	0006A	MOVL	#1, R0	0392
			04	0006D	RET		0393

; Routine Size: 110 bytes, Routine Base: \$CODE\$ + 00B9


```
0394 1 GLOBAL ROUTINE MAIL$CMD_SET_FORWARD (DELETE_FLAG) =
0395 1 ++
0396 1 FUNCTIONAL DESCRIPTION:
0397 1
0398 1     This routine is called to modify the forwarding address of a user.
0399 1
0400 1 INPUTS:
0401 1
0402 1     DELETE_FLAG Present if SET NOFORWARD. Parameter absent if SET FORWARD
0403 1     MAIL$Q_USERNAME Descriptor of username string
0404 1
0405 1 IMPLICIT INPUTS:
0406 1
0407 1     MAIL$T_SYSMAIRAB RAB for I/O on SYS$SYSTEM:VMSMAIL.DAT
0408 1
0409 1 OUTPUTS:
0410 1
0411 1     SYS$SYSTEM:VMSMAIL.DAT is created if it does not exist. The user
0412 1     record is modified, or added if it does not exist.
0413 1
0414 1 --
0415 2 BEGIN
0416 2
0417 2 BUILTIN
0418 2 NULLPARAMETER;
0419 2
0420 2 LOCAL
0421 2     NAMBUF : $BBLOCK[NAM$C_MAXRSS],
0422 2     DIRBUF : $BBLOCK[NAM$C_MAXRSS],
0423 2     TMPBUF : $BBLOCK[VMD_C_LENGTH + 3*NAM$C_MAXRSS],
0424 2     PTR,
0425 2     PROXY_FLAG,
0426 2     STATUS;
0427 2
0428 2 PROXY_FLAG = 0; ! Assume for self
0429 2 IF NULLPARAMETER(1)
0430 2 THEN BEGIN
0431 2     CL$GET VALUE(MAIL$SD SETF1,COMMAND_DESC); !Get forwarding address
0432 2     MAIL$STRIP_QUOTES(COMMAND_DESC,COMMAND_DESC);
0433 2     MAIL$UPCASE_Q(COMMAND_DESC);
0434 2     END
0435 2 ELSE STR$FREE1_DX(COMMAND_DESC); ! Ensure 0 forwarding address
0436 2
0437 2 IF_ERR(CHECK_STRING(COMMAND_DESC,NAM$C_MAXRSS);,
0438 2     RETURN .STATUS);
0439 2
0440 2 IF NOT CL$PRESENT(MAIL$SD USER) ! If not /USER=, then do self
0441 2 THEN CH$MOVE(VMD_C_LENGTH+.MAIL$L_USR$SYSBUF[VMD_B_FWDLNG]
0442 2     +.MAIL$L_USR$SYSBUF[VMD_B_FNMLNG]
0443 2     +.MAIL$L_USR$SYSBUF[VMD_B_DIRLNG],
0444 2     .MAIL$L_USR$SYSBUF,TMPBUF)
0445 2 ELSE BEGIN
0446 2     ! Ensure user has SYSNAM in process privilege mask
0447 2
0448 2     IF NOT MAIL$FORM_PRIV_CHECK()
0449 2     THEN RETURN TRUE;
0450 2
```

```
310 0451
311 0452
312 0453
313 0454
314 0455
315 0456
316 0457
317 0458
318 0459
319 0460
320 0461
321 0462
322 0463
323 0464
324 0465
325 0466
326 0467
327 0468
328 0469
329 0470
330 0471
331 0472
332 0473
333 0474
334 0475
335 0476
336 0477
337 0478
338 0479
339 0480
340 0481
341 0482
342 0483
343 0484
344 0485
345 0486
346 0487

PROXY FLAG = CLISGET VALUE(MAIL$SD_USER,DYN_DESC);
CH$COPY(.DYN_DESC[DSC$W_LENGTH], ! Copy name from /USER=
.DYN_DESC[DSC$A_POINTER],
VMD_C_USERNAME,TMPBUF[VMD_T_USERNAME]);
MAILST_SYMAIRAB[RAB$K_KBF] = TMPBUF; ! Set to read the record
MAILST_SYMAIRAB[RAB$K_UBF] = TMPBUF;
MAILST_SYMAIRAB[RAB$V_NLK] = FALSE;
IF NOT $GET(RAB=MAILST_SYMAIRAB) ! If it doesn't exist, zero record
THEN CH$FILL(0,VMD_C_LENGTH+NAM$C_MAXRSS-$BYTEOFFSET(VMD_W_FLAGS),
TMPBUF[VMD_W_FLAGS]);
END;
! If trying to forward to self, then set no forwarding
CH$MOVE(.TMPBUF[VMD_B_FNMLNG], ! Save full user name
TMPBUF[VMD_T_FWDNAM]+.TMPBUF[VMD_B_FWDLNG],NAMBUF);
CH$MOVE(.TMPBUF[VMD_B_DIRLNG], ! Save mail dir spec
TMPBUF[VMD_T_FWDNAM]+.TMPBUF[VMD_B_FWDLNG]+.TMPBUF[VMD_B_FNMLNG],
DIRBUF);
IF NOT .PROXY FLAG
AND CH$EQ(VMD_C_USERNAME,TMPBUF[VMD_T_USERNAME],
.COMMAND_DESC[DSC$W_LENGTH],.COMMAND_DESC[DSC$A_POINTER], ' ')
THEN TMPBUF[VMD_B_FWDLNG] = 0
ELSE BEGIN
CH$MOVE(.COMMAND_DESC[DSC$W_LENGTH],
.COMMAND_DESC[DSC$A_POINTER],
TMPBUF[VMD_T_FWDNAM]);
TMPBUF[VMD_B_FWDLNG] = .COMMAND_DESC[DSC$W_LENGTH];
END;
PTR = CH$MOVE(.TMPBUF[VMD_B_FNMLNG],NAMBUF, ! Restore full user name
TMPBUF[VMD_T_FWDNAM]+.TMPBUF[VMD_B_FWDLNG]);
CH$MOVE(.TMPBUF[VMD_B_DIRLNG],DIRBUF,.PTR);
MAIL$UPDATE_SYMAI_REC(TMPBUF); ! Update the record
2 RETURN TRUE
1 END;
```

```
OFFC 00000
5B 00000000G 00 9E 00002
5A 00000000G 00 9E 00009
59 00000000G 00 9E 00010
58 00000000G 00 9E 00017
5E FABC CE 9E 0001E
57 D4 00023
6C 95 00025
05 13 00027
04 AC D5 00029
1F 12 0002C
58 DD 0002E 13:
00000000G 00 9F 00030
6A 02 FB 00036
```

.EXTRN SYS\$GET

```
.ENTRY MAIL$CMD_SET_FORWARD, Save R2,R3,R4,R5,R6,- : 0394
R7,R8,R9,R10,R11
MOVAB MAIL$SD_USER, R11
MOVAB CLISGET_VALUE, R10
MOVAB MAILST_SYMAIRAB+48, R9
MOVAB COMMAND_DESC, R8
MOVAB -1348(SP), SP
CLRL PROXY_FLAG
TSTB (AP)
BEQL 1$
TSTL 4(AP)
BNEQ 2$
PUSHL R8
PUSHAB MAIL$SD_SETF1
CALLS #2, CLISGET_VALUE
0428
0429
0431
```

				58	DD	00039	PUSHL	R8		0432
				58	DD	0003B	PUSHL	R8		
	FF0F	CF		02	FB	0003D	CALLS	#2, MAIL\$STRIP_QUOTES		
				58	DD	00042	PUSHL	R8		0433
	00000000G	00		01	FB	00044	CALLS	#1, MAIL\$UPCASE_Q		
				09	11	0004B	BRB	3\$		0429
				58	DD	0004D	PUSHL	R8		0435
	00000000G	00		01	FB	0004F	CALLS	#1, STR\$FREE1_DX		
		7E	FF	8F	9A	00056	MOVZBL	#255, -(SP)		0438
				58	DD	0005A	PUSHL	R8		
	FF31	CF		02	FB	0005C	CALLS	#2, CHECK_STRING		
		01		50	E8	00061	BLBS	STATUS, 4\$		
					04	00064	RET			
				58	DD	00065	PUSHL	R11		0440
	00000000G	00		01	FB	00067	CALLS	#1, CLIS\$PRESENT		
		21		50	E8	0006E	BLBS	R0, 5\$		
		50	00000000G	00	DO	00071	MOVL	MAIL\$L_USR\$SYSBUF, R0		0441
		51	43	A0	9A	00078	MOVZBL	67(R0), R1		0442
		52	42	A0	9A	0007C	MOVZBL	66(R0), R2		
		51		52	CO	00080	ADDL2	R2, R1		
		52	41	A0	9A	00083	MOVZBL	65(R0), R2		0443
		51	44	A241	9E	00087	MOVAB	68(R2)(R1), R1		
6E		60		51	28	0008C	MOVCS	R1, (R0), TMPBUF		0441
				44	11	00090	BRB	7\$		
	00000000G	00		00	FB	00092	CALLS	#0, MAIL\$FORW_PRIV_CHECK		0449
		03		50	E8	00099	BLBS	R0, 6\$		
				00A1	31	0009C	BRW	10\$		
			08	A8	9F	0009F	PUSHAB	DYN_DESC		0452
				58	DD	000A2	PUSHL	R11		
		6A		02	FB	000A4	CALLS	#2, CLIS\$GET_VALUE		
		57		50	DO	000A7	MOVL	R0, PROXY_FLAG		
1F		50	0C	A8	DO	000AA	MOVL	DYN_DESC+4, R0		0454
	20	60	08	A8	2C	000AE	MOVCS	DYN_DESC, (R0), #32, #31, TMPBUF		0455
				6E		000B4				
		69		6E	9E	000B5	MOVAB	TMPBUF, MAIL\$T_SY\$MAIRAB+48		0456
	F4	A9		6E	9E	000B8	MOVAB	TMPBUF, MAIL\$T_SY\$MAIRAB+36		0457
	D6	A9		10	8A	000BC	BICB2	#16, MAIL\$T_SY\$MAIRAB+6		0458
			00	A9	9F	000C0	PUSHAB	MAIL\$T_SY\$MAIRAB		0459
	00000000G	00		01	FB	000C3	CALLS	#1, SY\$GET		
		09		50	E8	000CA	BLBS	R0, 7\$		
0124	8F	00		00	2C	000CD	MOVCS	#0, (SP), #0, #292, TMPBUF+31		0461
		6E		1F	AE	000D4				
		50		42	AE	9A	MOVZBL	TMPBUF+66, R0		0466
		56		43	AE	9A	MOVZBL	TMPBUF+67, R6		0467
	FF00	CD	44	AE46	50	28	MOVCS	R0, TMPBUF+68(R6), NAMBUF		0466
				51	AE	9A	MOVZBL	TMPBUF+65, R1		0468
				50	AE	9A	MOVZBL	TMPBUF+66, R0		0469
				50	56	CO	ADDL2	R6, R0		
	FE00	CD	44	AE40	51	28	MOVCS	R1, TMPBUF+68(R0), DIRBUF		0468
				11	57	E8	BLBS	PROXY_FLAG, 8\$		0471
				50	A8	DO	MOVL	COMMAND_DESC+4, R0		0473
68		20		6E	1F	2D	CMPCS	#31, TMPBUF, #32, COMMAND_DESC, (R0)		0472
					60					
					05	12	BNEQ	8\$		
			43	AE	94	00108	CLRB	TMPBUF+67		0474
				10	11	0010B	BRB	9\$		
		56		68	3C	0010D	MOVZWL	COMMAND_DESC, R6		0476

MAIL\$SET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

K 9
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]SET.B32;1

Page 13
(5)

44	AE		50	04	A8	D0	00110	MOVL	COMMAND, DESC+4, R0	:	0477
			60		56	28	00114	MOVCL	R6, (R0), TMPBUF+68	:	0478
		43	AE		56	90	00119	MOVB	R6, TMPBUF+67	:	0479
			51	42	AE	9A	0011D	MOVZBL	TMPBUF+66, R1	:	0481
			50	43	AE	9A	00121	MOVZBL	TMPBUF+67, R0	:	0482
44	AE40	FF00	CD		51	28	00125	MOVCL	R1, NAMBUF, TMPBUF+68[R0]	:	
			50	41	AE	9A	0012D	MOVZBL	TMPBUF+65, R0	:	0483
63	FE00	CD			50	28	00131	MOVCL	R0, DIRBUF, (PTR)	:	
					5E	DD	00137	PUSHL	SP	:	0484
	00000000G	00			01	FB	00139	CALLS	#1, MAIL\$UPDATE_SYSMAI_REC	:	
		50			01	D0	00140	MOVL	#1, R0	:	0486
					04	00143	10\$:	RET		:	0487

; Routine Size: 324 bytes, Routine Base: \$CODE\$ + 0127

MAIL\$SET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

L 9
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]SET.832;1

Page 14
(6)

```
0488 1 GLOBAL ROUTINE MAIL$CMD_SET_NOFORWARD =
0489 1 ++
0490 1 FUNCTIONAL DESCRIPTION:
0491 1
0492 1 This routine is called to turn off forwarding for a user
0493 1
0494 1 --
0495 1
0496 1 RETURN MAIL$CMD_SET_FORWARD(1);
```

```
0000 00000
01 DD 00002
01 FB 00004
04 00009
```

FEB3 CF

```
.ENTRY MAIL$CMD_SET_NOFORWARD, Save nothing
PUSHL #1
CALLS #1, MAIL$CMD_SET_FORWARD
RET
```

```
: 0488
: 0496
:
```

; Routine Size: 10 bytes, Routine Base: \$CODE\$ + 026B

```
0497 1 GLOBAL ROUTINE MAIL$CMD_SET_SELF =
0498 1 ++
0499 1 FUNCTIONAL DESCRIPTION:
0500 1
0501 1 Process the SET COPY_SELF command, which sets the commands on
0502 1 which an automatic copy is sent to the sender.
0503 1
0504 1 --
0505 2 BEGIN
0506 2 LOCAL
0507 2 DESC : VECTOR[2, LONG],
0508 2 STATUS;
0509 2
0510 2
0511 2 IF MAIL$GETLK_SYSMAL_REC(MAIL$Q_USERNAME, DESC)
0512 2 THEN CH$MOVE(.DESC[0], .DESC[1], MAIL$L_USRSYSBUF);
0513 2
0514 2 IF CL$PRESENT(SD_SEND)
0515 2 THEN MAIL$L_USRSYSBUF[VMD_V_SELFSEND] = 1
0516 2 ELSE IF CL$PRESENT(SD_NOSEND)
0517 2 THEN MAIL$L_USRSYSBUF[VMD_V_SELFSEND] = 0;
0518 2 IF CL$PRESENT(SD_REPLY)
0519 2 THEN MAIL$L_USRSYSBUF[VMD_V_SELFREPLY] = 1
0520 2 ELSE IF CL$PRESENT(SD_NOREPLY)
0521 2 THEN MAIL$L_USRSYSBUF[VMD_V_SELFREPLY] = 0;
0522 2
0523 2 RETURN MAIL$UPDATE_SYSMAL_REC(.MAIL$L_USRSYSBUF)
0524 1 END;
```

01FC 00000				.ENTRY	MAIL\$CMD_SET_SELF, Save R2,R3,R4,R5,R6,R7,-	
					R8	0497
	58	00000000G	00 9E 00002	MOVAB	CL\$PRESENT, R8	
	57	FDCA	CF 9E 00009	MOVAB	SD_SEND, R7	
	56	00000000G	00 9E 0000E	MOVAB	MAIL\$L_USRSYSBUF, R6	
	5E		08 C2 00015	SUBL2	#8, SP	
			5E DD 00018	PUSHL	SP	0511
		00000000G	00 9F 0001A	PUSHAB	MAIL\$Q_USERNAME	
	00000000G	00	02 FB 00020	CALLS	#2, MAIL\$GETLK_SYSMAL_REC	
	08		50 E9 00027	BLBC	R0, 1\$	
	50		66 D0 0002A	MOVL	MAIL\$L_USRSYSBUF, R0	0512
60	04	BE	6E 28 0002D	MOVCB	DESC, @DESC+4, (R0)	
			57 DD 00032	PUSHL	R7	0514
	68		01 FB 00034	CALLS	#1, CL\$PRESENT	
	09		50 E9 00037	BLBC	R0, 2\$	
	50		66 D0 0003A	MOVL	MAIL\$L_USRSYSBUF, R0	0515
	1F	A0	01 88 0003D	BISB2	#1, 31(R0)	
			10 11 00041	BRB	3\$	
		E4	A7 9F 00043	PUSHAB	SD_NOSEND	0516
	68		01 FB 00046	CALLS	#1, CL\$PRESENT	
	07		50 E9 00049	BLBC	R0, 3\$	
	50		66 D0 0004C	MOVL	MAIL\$L_USRSYSBUF, R0	0517
	1F	A0	01 8A 0004F	BICB2	#1, 31(R0)	
		10	A7 9F 00053	PUSHAB	SD_REPLY	0518

MAIL\$SET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

N 9
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]SET.B32;1

Page 16
(7)

	68		01	FB	00056	CALLS	#1, CL\$PRESENT	:	
	09		50	E9	00059	BLBC	RO, 4\$:	
	50		66	D0	0005C	MOVL	MAIL\$L USRSYSBUF, RO	:	0519
1F	A0		02	88	0005F	BISB2	#2, 31(RO)	:	
			10	11	00063	BRB	5\$:	
		F4	A7	9F	00065	PUSHAB	SD_NOREPLY	:	0520
	68		01	FB	00068	CALLS	#1, CL\$PRESENT	:	
	07		50	E9	00068	BLBC	RO, 5\$:	
	50		66	D0	0006E	MOVL	MAIL\$L USRSYSBUF, RO	:	0521
1F	A0		02	8A	00071	BICB2	#2, 31(RO)	:	
			66	DD	00075	PUSHL	MAIL\$L USRSYSBUF	:	0523
00000000G	00		01	FB	00077	CALLS	#1, MAIL\$UPDATE_SYSMAI_REC	:	
			04	0007E	RET			:	0524

; Routine Size: 127 bytes, Routine Base: \$CODE\$ + 0275

```
387 0525 1 GLOBAL ROUTINE MAIL$SET_NAME (FULL_DESC) =
388 0526 1 ++
389 0527 1 FUNCTIONAL DESCRIPTION:
390 0528 1
391 0529 1 This routine updates the string MAIL$Q_FULLUSER with the username
392 0530 1 from MAIL$Q_USERNAME and the full name passed in FULL_DESC
393 0531 1
394 0532 1 INPUTS:
395 0533 1
396 0534 1 FULL_DESC = address of descriptor of full name (without quotes)
397 0535 1
398 0536 1 OUTPUTS:
399 0537 1
400 0538 1 MAIL$Q_FULLUSER updated
401 0539 1 --
402 0540 2 BEGIN
403 0541 2
404 0542 2 MAP
405 0543 2 FULL_DESC : REF $BBLOCK;
406 0544 2
407 0545 2 BIND
408 0546 2 SPACE_DESC = $DESCRIPTOR(' ');
409 0547 2 QUOTE_DESC = $DESCRIPTOR('\"');
410 0548 2
411 0549 2 IF .FULL_DESC[DSC$W_LENGTH] NEQ 0
412 0550 2 THEN RETURN STR$CONCAT(MAIL$Q_FULLUSER,MAIL$Q_USERNAME,
413 0551 2 SPACE_DESC,QUOTE_DESC,.FULL_DESC,QUOTE_DESC)
414 0552 2 ELSE RETURN STR$COPY_DX(MAIL$Q_FULLUSER,MAIL$Q_USERNAME)
415 0553 1 END;
```

```
20 002F4 P.AAR: .ASCII \ \
00000001 002F5 .BLKB 3
00000000 002F8 P.AAQ: .LONG 1
00000000 002FC .ADDRESS P.AAR
22 00300 P.AAT: .ASCII \"
00000001 00301 .BLKB 3
00000001 00304 P.AAS: .LONG 1
00000000 00308 .ADDRESS P.AAT
```

```
SPACE_DESC= P.AAQ
QUOTE_DESC= P.AAS
```

```
54 00000000G 00 001C 00000 .ENTRY MAIL$SET NAME, Save R2,R3,R4
53 00000000G 00 9E 00002 MOVAB MAIL$Q_FULLUSER, R4
52 E5 AF 9E 00010 MOVAB MAIL$Q_USERNAME, R3
04 BC B5 00014 MOVAB QUOTE_DESC, R2
16 13 00017 TSTW @FULL_DESC
52 DD 00019 BEQL 1$
04 AC DD 0001B PUSHL R2
52 DD 0001E PUSHL FULL_DESC
F4 A2 9F 00020 PUSHL R2
53 DD 00023 PUSHAB SPACE_DESC
54 DD 00025 PUSHL R3
PUSHL R4
```

0525

0549

0550

0551

0550

MAIL\$SET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

C 10
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]SET.B32;1

Page 18
(8)

00000000G 00

06 FB 00027
04 0002E
53 DD 0002F 1\$:
54 DD 00031
02 FB 00033
04 0003A

CALLS #6, STR\$CONCAT
RET
PUSHL R3
PUSHL R4
CALLS #2, STR\$COPY_DX
RET

: 0552

: 0553

; Routine Size: 59 bytes, Routine Base: \$CODE\$ + 030C


```
417 0554 1 ROUTINE ALPHA_CHAR (CHAR) =
418 0555 1 ++
419 0556 1 FUNCTIONAL DESCRIPTION:
420 0557 1
421 0558 1 Tests the character. Returns true if A-Z or a-z, or $ or _, else error
422 0559 1 --
423 0560 2 BEGIN
424 0561 2 MAP
425 0562 2 CHAR : BYTE;
426 0563 2
427 0564 2 IF ((.CHAR GEQU %C'A') AND (.CHAR LEQU %C'Z'))
428 0565 2 OR ((.CHAR GEQU %C'a') AND (.CHAR LEQU %C'z'))
429 0566 2 OR (.CHAR EQL %C'$') OR (.CHAR EQL %C'_')
430 0567 2 THEN RETURN TRUE
431 0568 2 ELSE RETURN FALSE
432 0569 1 END;
```

```
0000 00000 ALPHA_CHAR:
      50      04 AC 9A 00002 .WORD Save nothing
41 8F      50 91 00006 MOVZBL CHAR, R0
      06 1F 0000A CMPB R0, #65
5A 8F      50 91 0000C BLSSU 1$
      17 1B 00010 CMPB R0, #90
61 8F      50 91 00012 1$: BLEQU 3$
      06 1F 00016 CMPB R0, #97
7A 8F      50 91 00018 BLSSU 2$
      0B 1B 0001C CMPB R0, #122
      24      50 91 0001E 2$: BLEQU 3$
      06 13 00021 BEQL 3$
5F 8F      50 91 00023 CMPB R0, #95
      04 12 00027 BNEQ 4$
      50      01 D0 00029 3$: MOVL #1, R0
      04 0002C RET
      50 D4 0002D 4$: CLRL R0
      04 0002F RET
```

```
0554
0564
0565
0566
0568
0569
```

; Routine Size: 48 bytes, Routine Base: \$CODE\$ + 0347

```
434 0570 1 ROUTINE SET_NAME (NAME_DESC) =
435 0571 1 |++
436 0572 1 | FUNCTIONAL DESCRIPTION:
437 0573 1 |
438 0574 1 |     Set the full username for the SET PERSONAL_NAME command
439 0575 1 |
440 0576 1 | --
441 0577 2 BEGIN
442 0578 2
443 0579 2 MAP
444 0580 2     NAME_DESC : REF $BBLOCK;
445 0581 2
446 0582 2 LOCAL
447 0583 2     DESC : VECTOR[2, LONG],
448 0584 2     DIRBUF : $BBLOCK[NAM$C_MAXRSS],
449 0585 2     TMPBUF : $BBLOCK[VMD_C_LENGTH + 3*NAM$C_MAXRSS],
450 0586 2     PTR : REF VECTOR[BYTE],
451 0587 2     STATUS;
452 0588 2
453 0589 2 MAIL$STRIP_QUOTES(.NAME_DESC, .NAME_DESC);
454 0590 2
455 P 0591 2 IF_ERR(CHECK_STRING(.NAME_DESC, 127));
456 0592 2     RETURN .STATUS;
457 0593 2
458 0594 2 IF .NAME_DESC[DSC$W_LENGTH] GTRU 0
459 0595 2 THEN BEGIN
460 0596 2     PTR = .NAME_DESC[DSC$A_POINTER];
461 0597 2     IF NOT ALPHA_CHAR(PTR[0])
462 0598 2         OR NOT CH$FAIL(CH$FIND_SUB(.NAME_DESC[DSC$W_LENGTH],
463 0599 2             .NAME_DESC[DSC$A_POINTER], 2, UPLIT(' ')))
464 0600 2         THEN RETURN SIGNAL(MAIL$_ILLPERNAM, 1, .NAME_DESC);
465 0601 2     END;
466 0602 2 MAIL$SET_NAME(.NAME_DESC);           ! Set new full name
467 0603 2
468 0604 2 MAIL$GETLK_SYSM$REC(MAIL$Q_USERNAME, DESC);
469 0605 2 CH$MOVE(.DESC[0], .DESC[1], .MAIL$Q_USR$SYSBUF);
470 0606 2 CH$MOVE(.MAIL$Q_USR$SYSBUF[VMD_B_DIRLNG],
471 0607 2     .MAIL$Q_USR$SYSBUF[VMD_T_FWDLNG] + .MAIL$Q_USR$SYSBUF[VMD_B_FWDLNG] +
472 0608 2     .MAIL$Q_USR$SYSBUF[VMD_B_FNLNG], DIRBUF);
473 0609 2 PTR = .MAIL$Q_USR$SYSBUF[VMD_T_FWDLNG] + .MAIL$Q_USR$SYSBUF[VMD_B_FWDLNG];
474 0610 2 .MAIL$Q_USR$SYSBUF[VMD_B_FNLNG] = .NAME_DESC[DSC$W_LENGTH];
475 0611 2 PTR = CH$MOVE(.NAME_DESC[DSC$W_LENGTH], .NAME_DESC[DSC$A_POINTER], PTR[0]);
476 0612 2 CH$MOVE(.MAIL$Q_USR$SYSBUF[VMD_B_DIRLNG], DIRBUF, PTR[0]);
477 0613 2
478 0614 2 RETURN MAIL$UPDATE_SYSM$REC(.MAIL$Q_USR$SYSBUF)
479 0615 1 END;
```

```
00 00 20 20 00377 .BLKB 1
00378 P.AAU: .ASCII \ \<0><0>
```

```
07FC 00000 SET_NAME:
SA FCF6 CF 9E C0002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10
MOVAB MAIL$STRIP_QUOTES, R10
```

```
: 0570
:
```

		5E	FBB4	CE	9E	00007	MOVAB	-1100(SP), SP		
		58	04	AC	DD	0000C	MOVL	NAME_DESC, R8	0589	
				58	DD	00010	PUSHL	R8		
				58	DD	00012	PUSHL	R8		
		6A		02	FB	00014	CALLS	#2, MAIL\$STRIP_QUOTES		
		7E	7F	8F	9A	00017	MOVZBL	#127, -(SP)	0592	
				58	DD	0001B	PUSHL	R8		
	41	AA		02	FB	0001D	CALLS	#2, CHECK_STRING		
		01		50	EB	00021	BLBS	STATUS, 1\$		
					04	00024	RET			
				68	B5	00025	TSTW	(R8)	0594	
				32	13	00027	BEQL	4\$		
		59	04	A8	DD	00029	MOVL	4(R8), PTR	0596	
		7E		69	9A	0002D	MOVZBL	(PTR), -(SP)	0597	
	02CF	CA		01	FB	00030	CALLS	#1, ALPHA_CHAR		
		11		50	E9	00035	BLBC	R0, 3\$		
04	BB		68	02	39	00038	MATCHC	#2, P.AAU, (R8), 24(R8)	0599	
		AF		03	13	0003F	BEQL	2\$		
		53		02	DD	00041	MOVL	#2, R3		
		53		02	C2	00044	SUBL2	#2, R3		
				12	13	00047	BEQL	4\$		
				58	DD	00049	PUSHL	R8	0600	
				01	DD	0004B	PUSHL	#1		
				8F	DD	0004D	PUSHL	#8290554		
	00000000G	00		03	FB	00053	CALLS	#3, LIB\$SIGNAL		
					04	0005A	RET			
				58	DD	0005B	PUSHL	R8	0602	
	0294	CA		01	FB	0005D	CALLS	#1, MAIL\$SET_NAME		
				F8	AD	9F	PUSHAB	DESC	0604	
				00000000G	00	9F	PUSHAB	MAIL\$Q USERNAME		
				02	FB	0006B	CALLS	#2, MAIL\$GETLK_SYSMAL_REC		
	00000000G	00		00	DD	00072	MOVL	MAIL\$L USRSYSBUF, R6	0605	
66		FC	BD	F8	AD	28	MOVC3	DESC, 2DESC+4, (R6)		
			51	41	A6	9A	MOVZBL	65(R6), R1	0606	
			50	43	A6	9A	MOVZBL	67(R6), R0	0607	
57			56		50	C1	ADDL3	R0, R6, R7		
			50		A6	9A	MOVZBL	66(R6), R0	0608	
FEF8	CD	44	A047		51	28	MOVC3	R1, 68(R0)[R7], DIRBUF	0606	
			59		A7	9E	MOVAB	68(R7), PTR	0609	
		42	A6		68	90	MOVB	(R8), 66(R6)	0610	
69		04	B8		68	28	MOVC3	(R8), 24(R8), (PTR)	0611	
			59		53	DD	MOVL	R3, PTR		
			50	41	A6	9A	MOVZBL	65(R6), R0	0612	
69	FEF8	CD			50	28	MOVC3	R0, DIRBUF, (PTR)		
					56	DD	PUSHL	R6	0614	
	00000000G	00			01	FB	CALLS	#1, MAIL\$UPDATE_SYSMAL_REC		
					04	000BA	RET		0615	

; Routine Size: 187 bytes, Routine Base: \$CODE\$ + 037C

```
.. 481      0616 1 GLOBAL ROUTINE MAIL$CMD_SET_NAME =  
.. 482      0617 1 |**  
.. 483      0618 1 | FUNCTIONAL DESCRIPTION:  
.. 484      0619 1 |  
.. 485      0620 1 |         Process the SET PERSONAL_NAME command  
.. 486      0621 1 | --  
.. 487      0622 2 BEGIN  
.. 488      0623 2  
.. 489      0624 2 CLISGET VALUE(MAIL$SD SETN1,COMMAND_DESC);!Get full username  
.. 490      0625 2 IF .COMMAND_DESC(DSC$D_LENGTH) EQL 0  
.. 491      0626 2     THEN RETURN TRUE;  
.. 492      0627 2  
.. 493      0628 2 RETURN SET_NAME(COMMAND_DESC)  
.. 494      0629 1 END;
```

```
                                0004 00000  
                                00 9E 00002  
52 00000000' 00 52 DD 00009  
                                00 9F 0000B  
00000000G 00 02 FB 00011  
                                62 B5 00018  
                                04 12 0001A  
50 01 D0 0001C  
                                04 0001F  
                                52 DD 00020 1$:  
FF1E CF 01 FB 00022  
                                04 00027
```

```
.ENTRY MAIL$CMD SET_NAME, Save R2  
MOVAB  COMMAND_DESC, R2  
PUSHL  R2  
PUSHAB MAIL$SD SETN1  
CALLS  #2, CLISGET VALUE  
TSTW   COMMAND_DESC  
BNEQ   1$  
MOVL   #1, R0  
RET  
PUSHL  R2  
CALLS  #1, SET_NAME  
RET
```

```
: 0616  
: 0624  
: 0625  
: 0626  
: 0628  
: 0629
```

; Routine Size: 40 bytes, Routine Base: \$CODE\$ + 0437

MAIL\$SET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

H 10
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]SET.B32;1

Page 23
(12)

```

: 496      0630 1 GLOBAL ROUTINE MAIL$CMD_CLEAR_NAME =
: 497      0631 1 ++
: 498      0632 1 FUNCTIONAL DESCRIPTION:
: 499      0633 1
: 500      0634 1 Process the SET NOPERSONAL_NAME command (clear the personal name)
: 501      0635 1
: 502      0636 1 --
: 503      0637 2 BEGIN
: 504      0638 2
: 505      0639 2 LOCAL
: 506      0640 2 DESC : VECTOR[2, LONG];
: 507      0641 2
: 508      0642 2 DESC[0] = 0;
: 509      0643 2 DESC[1] = DESC[0]; !Point to valid address
: 510      0644 2 RETURN SET_NAME(DESC)
: 511      0645 1 END;
```

			0000 00000	.ENTRY MAIL\$CMD_CLEAR_NAME, Save nothing	: 0630
	5E	04	C2 00002	SUBL2 #4, SP	: 0642
		7E	D4 00005	CLRL DESC	: 0643
04	AE	6E	9E 00007	MOVAB DESC, DESC+4	: 0644
		5E	DD 0000B	PUSHL SP	: 0645
FF0B	CF	01	FB 0000D	CALLS #1, SET_NAME	
			04 00012	RET	

; Routine Size: 19 bytes, Routine Base: \$CODE\$ + 045F

```
0646 1 GLOBAL ROUTINE MAIL$CMD_SET_AUTO =
0647 1 ++
0648 1 FUNCTIONAL DESCRIPTION:
0649 1
0650 1 Do the SET AUTO_PURGE command
0651 1 --
0652 2 BEGIN
0653 2
0654 2 LOCAL
0655 2 DESC : VECTOR[2, LONG];
0656 2
P 0657 2 IF_ERR(MAIL$GETLK_SYSMAL_REC(MAIL$Q_USERNAME, DESC);,
0658 2 RETURN TRUE);
0659 2 CH$MOVE(.DESC[0], .DESC[1], .MAIL$L_USRSYSBUF);
0660 2 MAIL$L_USRSYSBUF[VMD V NOAUTOPURGE] = 0;
0661 2 RETURN MAIL$UPDATE_SYSMAL_REC(.MAIL$L_USRSYSBUF)
0662 1 END;
```

			007C 00000	.ENTRY MAIL\$CMD_SET_AUTO, Save R2,R3,R4,R5,R6	: 0646
	5E		08 C2 00002	SUBL2 #8, SP	
			5E DD 00005	PUSHL SP	: 0658
		00000000G	00 9F 00007	PUSHAB MAIL\$Q_USERNAME	
	00000000G	00	02 FB 0000D	CALLS #2, MAIL\$GETLK_SYSMAL_REC	
		04	50 E8 00014	BLBS STATUS, 1\$	
		50	01 D0 00017	MOVL #1, R0	
			04 0001A	RET	
		56 00000000G	00 D0 0001B 1\$:	MOVL MAIL\$L_USRSYSBUF, R6	: 0659
66	04	BE	6E 28 00022	MOVC3 DESC, 3DESC+4, (R6)	
	1F	A6	04 8A 00027	BICB2 #4, 31(R6)	: 0660
			56 DD 0002B	PUSHL R6	: 0661
	00000000G	00	01 FB 0002D	CALLS #1, MAIL\$UPDATE_SYSMAL_REC	
			04 00034	RET	: 0662

; Routine Size: 53 bytes. Routine Base: \$CODE\$ + 0472

```
0663 1 GLOBAL ROUTINE MAIL$CMD_SET_NOAUTO =
0664 1 ++
0665 1 FUNCTIONAL DESCRIPTION:
0666 1
0667 1 Do the SET AUTO_PURGE command
0668 1 --
0669 2 BEGIN
0670 2 LOCAL
0671 2 DESC : VECTOR[2, LONG];
0672 2
0673 2 IF_ERR(MAIL$GETLK_SYSMAL_REC(MAIL$Q_USERNAME, DESC);,
0674 2 RETURN TRUE);
0675 2 CH$MOVE(.DESC[0], .DESC[1], .MAIL$L_USRSYSBUF);
0676 2
0677 2 MAIL$L_USRSYSBUF[VMD_V_NOAUTOPURGE] = 1;
0678 2
0679 2 RETURN MAIL$UPDATE_SYSMAL_REC(.MAIL$L_USRSYSBUF)
0680 1 END;
```

```
007C 00000
5E 08 C2 00002
00000000G 5E DD 00005
00000000G 00 00 9F 00007
04 02 FB 0000D
50 50 E8 00014
56 01 D0 00017
00000000G 00 04 0001A
66 04 56 00000000G 00 D0 0001B 1$:
1F BE 6E 28 00022
A6 04 88 00027
00000000G 00 56 DD 0002B
01 FB 0002D
04 00034
```

```
.ENTRY MAIL$CMD_SET_NOAUTO, Save R2,R3,R4,R5,R6 : 0663
SUBL2 #8, SP : 0674
PUSHL SP
PUSHAB MAIL$Q_USERNAME
CALLS #2, MAIL$GETLK_SYSMAL_REC
BLBS STATUS, 1$
MOVL #1, R0
RET
MOVL MAIL$L_USRSYSBUF, R6 : 0675
MOVC3 DESC, 3DESC+4, (R6)
BISB2 #4, 31(R6) : 0677
PUSHL R6 : 0679
CALLS #1, MAIL$UPDATE_SYSMAL_REC
RET : 0680
```

; Routine Size: 53 bytes. Routine Base: \$CODE\$ + 04A7

```
550 0681 1 GLOBAL ROUTINE MAIL$CMD_SET_WASTE =
551 0682 1 ++
552 0683 1 FUNCTIONAL DESCRIPTION:
553 0684 1
554 0685 1 Process the SET WASTEBASKET_FOLDER command
555 0686 1
556 0687 2 BEGIN
557 0688 2
558 0689 2 LOCAL
559 0690 2 RECDISC : VECTOR[2, LONG],
560 0691 2 RECBUF : $BLOCK[$BYTEOFFSET(MFR_T_DATA)+MFR_S_FILEKEY+1];
561 0692 2
562 0693 2 BIND
563 0694 2 RECORD TYPE = RECBUF[MFR_Q_DATUM] : VECTOR[ LONG],
564 0695 2 WASTENAME = RECBUF[MFR_T_DATA] : VECTOR[ BYTE];
565 0696 2
566 0697 2 CLISGET VALUE(SD WASTENAME, COMMAND_DESC);
567 0698 2 IF .COMMAND_DESC[DSCSW_LENGTH] EQL 0
568 0699 2 THEN RETURN TRUE;
569 0700 2
570 P 0701 2 IF_ERR(MAIL$OPEN IF());,
571 0702 2 RETURN TRUE;
572 0703 2
573 0704 2 IF NOT .MAIL$GL_FLAGS[MAIF V ISAM]
574 0705 2 THEN RETURN SIGNAL(MAIL$NOTISAM);
575 0706 2
576 0707 2 IF CH$EQL(.COMMAND_DESC[DSCSW_LENGTH], .COMMAND_DESC[DSCSA_POINTER],
577 0708 2 .MAIL$SD_MAIL[DSCSW_LENGTH], .MAIL$SD_MAIL[DSCSA_POINTER])
578 0709 2 OR CH$EQL(.COMMAND_DESC[DSCSW_LENGTH], .COMMAND_DESC[DSCSA_POINTER],
579 0710 2 .MAIL$SD_NEWMAIL[DSCSW_LENGTH], .MAIL$SD_NEWMAIL[DSCSA_POINTER])
580 0711 2 THEN RETURN SIGNAL(MAIL$ILLFOLNAM, 1, COMMAND_DESC);
581 0712 2
582 0713 2 CH$FILL(0, $BYTEOFFSET(MFR_T_DATA), RECBUF);
583 0714 2 RECORD TYPE[0] = MFR_C_WASTENAME;
584 0715 2 WASTENAME[0] = .COMMAND_DESC[DSCSW_LENGTH];
585 0716 2 CH$MOVE(.COMMAND_DESC[DSCSW_LENGTH], .COMMAND_DESC[DSCSA_POINTER],
586 0717 2 WASTENAME[1]);
587 0718 2
588 0719 2 RECDISC[0] = $BYTEOFFSET(MFR_T_DATA) + .COMMAND_DESC[DSCSW_LENGTH] + 1;
589 0720 2 RECDISC[1] = RECBUF;
590 0721 2 IF MAIL$UPDATE_INFO_REC(MAIL$T_RD_RAB, RECDISC)
591 0722 2 THEN BEGIN
592 0723 2
593 0724 2 Copy anything in old wastebasket folder to new wastebasket folder
594 0725 2
595 0726 2 MAIL$MOVE_WASTEBASKET(COMMAND_DESC);
596 0727 2 STR$COPY_DX(MAIL$Q_WASTEDESC, COMMAND_DESC);
597 0728 2 END;
598 0729 2 RETURN TRUE
599 0730 1 END;
```

01FC 00000

.ENTRY MAIL\$CMD_SET_WASTE, Save R2,R3,R4,R5,R6,R7,-; 0681
R8

		58	00000000G	00	9E	00002	MOVAB	LIB\$SIGNAL, R8	
		57	000000000	00	9E	00009	MOVAB	COMMAND_DESC, R7	
		5E	AO	AE	9E	00010	MOVAB	-96(SP), SP	
				57	DD	00014	PUSHL	R7	0697
			FB7A	CF	9F	00016	PUSHAB	SD_WASTENAME	
	00000000G	00		02	FB	0001A	CALLS	#2, CL\$GET_VALUE	0698
				67	B5	00021	TSTW	COMMAND_DESC	
				03	12	00023	BNEQ	2\$	
				00A3	31	00025	BRW	6\$	
	00000000G	00		00	FB	00028	CALLS	#0, MAIL\$OPEN_IF	0702
		F3		50	E9	0002F	BLBC	STATUS, 1\$	
		0A	00000000G	00	E8	00032	BLBS	MAIL\$GL_FLAGS, 3\$	0704
			007E80AA	8F	DD	00039	PUSHL	#829047\$	0705
		68		01	FB	0003F	CALLS	#1, LIB\$SIGNAL	
					04	00042	RET		
		55		67	3C	00043	MOVZWL	COMMAND_DESC, R5	0707
		54	04	A7	D0	00046	MOVL	COMMAND_DESC+4, R4	
		50	00000000G	00	D0	0004A	MOVL	MAIL\$SD_MAIL+4, R0	0708
00000000G	00			55	2D	00051	CMPC5	R5, (R4), #0, MAIL\$SD_MAIL, (R0)	0707
		64		60		0005A			
				13	13	0005B	BEQL	4\$	
		50	00000000G	00	D0	0005D	MOVL	MAIL\$SD_NEWMAIL+4, R0	0710
00000000G	00			55	2D	00064	CMPC5	R5, (R4), #0, MAIL\$SD_NEWMAIL, (R0)	0709
		64		60		0006D			
				0E	12	0006E	BNEQ	5\$	
				57	DD	00070	PUSHL	R7	0711
				01	DD	00072	PUSHL	#1	
			007E80E2	8F	DD	00074	PUSHL	#8290530	
		68		03	FB	0007A	CALLS	#3, LIB\$SIGNAL	
					04	0007D	RET		
	30			00	2C	0007E	MOVCS	#0, (SP), #0, #48, RECBUF	0713
		6E		6E		00083			
		6E		02	D0	00084	MOVL	#2, RECORD TYPE	0714
		56		67	3C	00087	MOVZWL	COMMAND_DESC, R6	0715
		AE	30	56	90	0008A	MOVB	R6, WASTENAME	
		50		A7	D0	0008E	MOVL	COMMAND_DESC+4, R0	0716
		60	04	56	28	00092	MOVCS	R6, (R0), WASTENAME+1	0717
31	AE			A6	9E	00097	MOVAB	49(R6), RECDISC	0719
		58		6E	9E	0009C	MOVAB	RECBUF, RECDISC+4	0720
		5C		AE	9F	000A0	PUSHAB	RECDISC	0721
			58	AE	9F	000A3	PUSHAB	MAIL\$T_RD_RAB	
			00000000G	00	FB	000A9	CALLS	#2, MAIL\$UPDATE_INFO_REC	
				02	FB	000A9	CALLS	#2, MAIL\$UPDATE_INFO_REC	
				50	E9	000B0	BLBC	R0, 6\$	
				57	DD	000B3	PUSHL	R7	0726
				01	FB	000B5	CALLS	#1, MAIL\$MOVE_WASTEBASKET	
				57	DD	000BC	PUSHL	R7	0727
			00000000G	00	9F	000BE	PUSHAB	MAIL\$Q_WASTEDESC	
				02	FB	000C4	CALLS	#2, STR\$COPY_DX	
				01	D0	000CB	MOVL	#1, R0	0729
				04	000CE	RET			0730

; Routine Size: 207 bytes, Routine Base: \$CODE\$ + 04DC

MAIL\$SET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

M 10
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]SET.B32;1

Page 28
(16)

```

: 601      0731 1 GLOBAL ROUTINE MAIL$CMD_SET_FILE =
: 602      0732 1 ++
: 603      0733 1 FUNCTIONAL DESCRIPTION:
: 604      0734 1
: 605      0735 1 process the SET FILE command
: 606      0736 1
: 607      0737 1 --
: 608      0738 2 BEGIN
: 609      0739 2 CLISGET_VALUE(SD_FILE,COMMAND_DESC);
: 610      0740 2
: 611      0741 2 MAIL$OPEN_FILE(COMMAND_DESC);
: 612      0742 2 RETURN TRUE
: 613      0743 1 END;
```

```

                                0004 00000
                                00 9E 00002
                                52 DD 00009
                                CF 9F 0000B
                                02 FB 0000F
                                52 DD 00016
                                01 FB 00018
                                01 D0 0001F
                                04 00022
52 00000000'
FA5A
00000000G 00
00000000G 00
50
```

```

.ENTRY MAIL$CMD_SET_FILE, Save R2
MOVAB  COMMAND_DESC, R2
PUSHL  R2
PUSHAB SD_FILE
CALLS  #2, CLISGET_VALUE
PUSHL  R2
CALLS  #1, MAIL$OPEN_FILE
MOVL   #1, R0
RET
```

```

: 0731
: 0739
: 0741
: 0742
: 0743
```

: Routine Size: 35 bytes, Routine Base: \$CODE\$ + 05AB

```
615 0744 1 ROUTINE RENAME_SUCCESS (OLD_SPEC,NEW_SPEC,LOG_FLAG) =
616 0745 1 |++
617 0746 1 | FUNCTIONAL DESCRIPTION:
618 0747 1 |
619 0748 1 | This routine is called by LIB$RENAME_FILE on a successful RENAME
620 0749 1 | --
621 0750 2 BEGIN
622 0751 2
623 0752 2 MAP
624 0753 2 OLD_SPEC : REF $BLOCK;
625 0754 2
626 0755 2 IF CH$EQL(.OLD_SPEC[DSC$W_LENGTH],.OLD_SPEC[DSC$A_POINTER],
627 0756 2 .MAIL$Q_RD_DESC[DSC$W_LENGTH],.MAIL$Q_RD_DESC[DSC$A_POINTER])
628 0757 2 THEN STR$COPY_DX(MAIL$Q_RD_DESC,.NEW_SPEC);
629 0758 2
630 0759 2 IF .LOG_FLAG
631 0760 2 THEN SIGNAL(MAIL$ RENAMED,2,.OLD_SPEC,.NEW_SPEC);
632 0761 2
633 0762 2 RETURN TRUE
634 0763 1 END;
```

				003C 00000 RENAME_SUCCESS:							
			55	00000000G	00	9E	00002	WORD	Save R2,R3,R4,R5		0744
			54	04	AC	D0	00009	MOVAB	MAIL\$Q_RD_DESC, R5		
			50	04	A5	D0	0000D	MOVL	OLD_SPEC, R4		0755
65	00	04	B4		64	2D	00011	MOVL	MAIL\$Q_RD_DESC+4, R0		0756
					60		00017	CMPC5	(R4), R4(R4), #0, MAIL\$Q_RD_DESC, (R0)		0755
					0C	12	00018	BNEQ	1\$		
				08	AC	DD	0001A	PUSHL	NEW_SPEC		0757
					55	DD	0001D	PUSHL	R5		
		00000000G	00		02	FB	0001F	CALLS	#2, STR\$COPY_DX		
			14		0C	AC	E9 00026	BLBC	LOG_FLAG, 2\$		0759
				08	AC	DD	0002A	PUSHL	NEW_SPEC		0760
					54	DD	0002D	PUSHL	R4		
					02	DD	0002F	PUSHL	#2		
				007E12A3	8F	DD	00031	PUSHL	#8262307		
		00000000G	00		04	FB	00037	CALLS	#4, LIB\$SIGNAL		
			50		01	D0	0003E	MOVL	#1, R0		0762
					04	00	0041	RET			0763

; Routine Size: 66 bytes, Routine Base: \$CODE\$ + 05CE

```
0764 1 ROUTINE RENAME_ERROR (OLD_SPEC,NEW_SPEC,RMSSTS,RMSSTV,ERROR_SOURCE,LOG_FLAG) =
0765 1 ++
0766 1 FUNCTIONAL DESCRIPTION:
0767 1
0768 1 Error routine called by LIB$RENAME_FILE on an error
0769 1 --
0770 2 BEGIN
0771 2 MAP
0772 2 RMSSTS : REF VECTOR[,LONG],
0773 2 RMSSTV : REF VECTOR[,LONG];
0774 2
0775 2 CASE ERROR_SOURCE FROM 0 TO 2
0776 2 OF SET
0777 2
0778 2 [0] : !Error searching for old_file
0779 2 IF RMSSTS[0] EQL RMS$_FNF
0780 2 THEN RETURN TRUE
0781 2 ELSE SIGNAL(MAIL$_SEARCHFAIL,1,OLD_SPEC
0782 2 RMSSTS[0],RMSSTV[0]);
0783 2
0784 2 [1,2] : !Error parsing new file
0785 2 SIGNAL(MAIL$_SEARCHFAIL,1,NEW_SPEC,RMSSTS[0],RMSSTV[0]);
0786 2 TES;
0787 2
0788 2 RETURN FALSE !Force LIB$RENAME_FILE to terminate
0789 1 END;
```

0000 00000 RENAME_ERROR:									
02	00	14	AC	CF	00002	1\$:	WORD	Save nothing	0764
001F	001F	0006	00007				CASEL	ERROR_SOURCE, #0, #2	0775
							WORD	2\$-1\$,-	
								4\$-1\$,-	
								4\$-1\$	
00018292	8F	0C	BC	D1	0000D	2\$:	CMPL	@RMSSTS, #98962	0779
	50		04	12	00015		BNEQ	3\$	
			01	DD	00017		MOVL	#1, R0	0780
				04	0001A		RET		
		10	BC	DD	0001B	3\$:	PUSHL	@RMSSTV	0782
		0C	BC	DD	0001E		PUSHL	@RMSSTS	
		04	AC	DD	00021		PUSHL	OLD_SPEC	0781
			09	11	00024		BRB	5\$	
		10	BC	DD	00026	4\$:	PUSHL	@RMSSTV	0785
		0C	BC	DD	00029		PUSHL	@RMSSTS	
		08	AC	DD	0002C		PUSHL	NEW_SPEC	
			01	DD	0002F	5\$:	PUSHL	#1	
00000000G	00	007E123A	8F	DD	00031		PUSHL	#8262202	
			05	FB	00037		CALLS	#5, LIB\$SIGNAL	
			50	D4	0003E		CLRL	R0	0788
			04	00040			RET		0789

; Routine Size: 65 bytes, Routine Base: \$CODE\$ + 0610


```
663 0790 1 ROUTINE SET_NEW_DIR (SUB_DESC, FULL_DESC, LOG_FLAG) =
664 0791 1 ++
665 0792 1 FUNCTIONAL DESCRIPTION:
666 0793 1
667 0794 1 Rename MAIL.MAI into the new directory. Then update
668 0795 1 VMSMAIL.DAT to have the new subdirectory spec. Then rename
669 0796 1 the rest of the .MAI files
670 0797 1
671 0798 1 Inputs:
672 0799 1
673 0800 1 SUB_DESC = address of descriptor of subdir ([.MAIL])
674 0801 1 FULL_DESC = address of descriptor of full subdir name (d:[user.mail])
675 0802 1 LOG_FLAG = true to /log, false otherwise
676 0803 1
677 0804 1 --
678 0805 2 BEGIN
679 0806 2
680 0807 2 MAP
681 0808 2 SUB_DESC : REF $BLOCK,
682 0809 2 FULL_DESC : REF $BLOCK;
683 0810 2
684 0811 2 LOCAL
685 0812 2 DESC : VECTOR[2, LONG];
686 0813 2
687 0814 2 Rename MAIL.MAI; * to the new directory
688 0815 2
689 0816 2 STR$CONCAT(DYN_DESC, MAIL$Q_MAIL_MAI, $DESCRIPTOR(';', '*'));
690 P 0817 2 IF_ERR(LIB$RENAME_FILE(DYN_DESC, FULL_DESC, MAIL$Q_USERDEF, 0, 0,
691 P 0818 2 RENAME_SUCCESS, RENAME_ERROR, 0, LOG_FLAG));
692 0819 2 RETURN TRUE);
693 0820 2
694 0821 2
695 0822 2 Update record in VMSMAIL. If an error occurs, rename MAIL.MAI; * to
696 0823 2 the old directory.
697 0824 2
698 P 0825 2 IF_ERR(MAIL$GETLK_SYSMAL_REC(MAIL$Q_USERNAME, DESC));
699 P 0826 2 LIB$RENAME_FILE(DYN_DESC, MAIL$Q_USERDEF, FULL_DESC, 0, 0,
700 P 0827 2 RENAME_SUCCESS, RENAME_ERROR, 0, 1); !a(ways log
701 0828 2 RETURN TRUE);
702 0829 2 CH$MOVE(.DESC[0], DESC[1], MAIL$Q_USR$SYSBUF);
703 0830 2 CH$MOVE(.SUB_DESC[DSC$W_LENGTH], SUB_DESC[DSC$A_POINTER],
704 0831 2 MAIL$Q_USR$SYSBUF[VMD_T_FWDNAM] + MAIL$Q_USR$SYSBUF[VMD_B_FWDLNG] +
705 0832 2 MAIL$Q_USR$SYSBUF[VMD_B_FNLNG]);
706 0833 2 MAIL$Q_USR$SYSBUF[VMD_B_DIRLNG] = .SUB_DESC[DSC$W_LENGTH];
707 0834 2
708 P 0835 2 IF_ERR(MAIL$UPDATE_SYSMAL_REC(.MAIL$Q_USR$SYSBUF));
709 P 0836 2 LIB$RENAME_FILE(DYN_DESC, MAIL$Q_USERDEF, FULL_DESC, 0, 0,
710 P 0837 2 RENAME_SUCCESS, RENAME_ERROR, 0, 1); !a(ways log
711 0838 2 RETURN TRUE);
712 0839 2
713 0840 2
714 0841 2 Finally rename the rest of the mail files to the new directory
715 0842 2
716 0843 2 LIB$RENAME_FILE($DESCRIPTOR('*.MAI; *'), FULL_DESC, MAIL$Q_USERDEF, 0, 0,
717 0844 2 RENAME_SUCCESS, RENAME_ERROR, 0, LOG_FLAG);
718 0845 2
719 0846 2 Set up the strings containing the mail file directory spec and the
```

```
0847 2 | default mail file spec.  
0848 2 |  
0849 2 | STR$COPY DX(MAIL$Q_USERDEF, FULL_DESC);  
0850 2 | STR$CONCAT(MAIL$Q_MAILDEF, FULL_DESC, MAIL$Q_MAIL_MAI);  
0851 2 | RETURN TRUE  
0852 1 | END;
```

```
2A 3B 49 41 4D 2E 2A 00651 P.AAW: .ASCII \;*\  
00000002 00653 .BLKB 1  
00000000 00654 P.AAV: .LONG 2  
00000000 00658 .ADDRESS P.AAW  
2A 3B 49 41 4D 2E 2A 0065C P.AAY: .ASCII \*.MAI;*\  
00000007 00663 .BLKB 1  
00000000 00664 P.AAX: .LONG 7  
00000000 00668 .ADDRESS P.AAY
```

```
OFFC 00000 SET_NEW_DIR:  
5B 00000000 00 9E 00002 .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11 0790  
5A 00000000G 00 9E 00009 MOVAB DYN_DESC, R11  
59 91 AF 9E 00010 MOVAB MAIL$Q_USERDEF, R10  
5E 08 C2 00014 MOVAB RENAME_ERROR, R9  
CE AF 9F 00017 SUBL2 #8, SP  
F97E CF 9F 0001A PUSHAB P.AAV 0816  
5B DD 0001E PUSHAB MAIL$Q_MAIL_MAI  
00000000G 00 03 FB 00020 PUSHL R11  
OC AC DD 00027 CALLS #3, STR$CONCAT 0819  
7E D4 0002A PUSHL LOG_FLAG  
59 DD 0002C CLRL -(SP)  
BE A9 9F 0002E PUSHL R9  
7E 7C 00031 PUSHAB RENAME_SUCCESS  
5A DD 00033 CLRL -(SP)  
58 08 AC DD 00035 PUSHL R10  
58 DD 00039 MOVL FULL_DESC, R8  
5B DD 0003B PUSHL R8  
00000000G 00 09 FB 0003D PUSHL R11  
5C 50 E9 00044 CALLS #9, LIB$RENAME_FILE  
5E DD 00047 BLBC STATUS, 2$ 0828  
00000000G 00 00 9F 00049 PUSHL SP  
00000000G 00 02 FB 0004F PUSHAB MAIL$Q_USERNAME  
32 50 E9 00056 CALLS #2, MAIL$GETLK_SYSMAL_REC  
56 00000000G 00 DD 00059 BLBC STATUS, 1$ 0829  
66 04 BE 6E 28 00060 MOVL MAIL$Q_USR$SYSBUF, R6  
57 04 AC DD 00065 MOVCL3 DESC, 3DESC+4, (R6) 0830  
50 43 A6 9A 00069 MOVL SUB_DESC, R7 0831  
50 56 C0 0006D MOVZBL 67(R6), R0  
51 42 A6 9A 00070 ADDL2 R6, R0 0832  
44 A140 04 B7 67 28 00074 MOVCL3 (R7), 64(R7), 68(R1)[R0] 0831  
41 A6 67 90 0007B MOVCL3 (R7), 65(R6) 0833  
00000000G 00 56 DD 0007F PUSHL R6 0838  
1A 01 FB 00081 CALLS #1, MAIL$UPDATE_SYSMAL_REC  
50 E8 00088 BLBS STATUS, 3$  
01 DD 0008B 1$: PUSHL #1
```

MAILSSET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

E 11
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]SET.B32;1

Page 33
(19)

		7E	D4	0008D	CLRL	-(SP)	
		59	DD	0008F	PUSHL	R9	
	BE	A9	9F	00091	PUSHAB	RENAME_SUCCESS	
		7E	7C	00094	CLRQ	-(SP)	
		58	DD	00096	PUSHL	R8	
		5A	DD	00098	PUSHL	R10	
		5B	DD	0009A	PUSHL	R11	
00000000G	00	09	FB	0009C	CALLS	#9, LIB\$RENAME_FILE	
		39	11	000A3	BRB	4\$	
	OC	AC	DD	000A5	PUSHL	LOG FLAG	0844
		7E	D4	000A8	CLRL	-(SP)	0843
		59	DD	000AA	PUSHL	R9	
	BE	A9	9F	000AC	PUSHAB	RENAME_SUCCESS	
		7E	7C	000AF	CLRQ	-(SP)	
	0500	8F	BB	000B1	PUSHR	#*M<R8,R10>	
	FF3F	CF	9F	000B5	PUSHAB	P.AAX	
00000000G	00	09	FB	000B9	CALLS	#9, LIB\$RENAME_FILE	0849
		58	DD	000C0	PUSHL	R8	
00000000G	00	5A	DD	000C2	PUSHL	R10	
		02	FB	000C4	CALLS	#2, STR\$COPY DX	
	F8CD	CF	9F	000CB	PUSHAB	MAIL\$Q_MAIL_MAI	0850
		58	DD	000CF	PUSHL	R8	
	00000000G	00	9F	000D1	PUSHAB	MAIL\$Q_MAILDEF	
00000000G	00	03	FB	000D7	CALLS	#3, STR\$CONCAT	
	50	01	DD	000DE	MOVL	#1, R0	0851
		04	000E1	RET			0852

; Routine Size: 226 bytes, Routine Base: \$CODE\$ + 066C

```
0853 1 GLOBAL ROUTINE MAIL$CMD_SET_DIR =
0854 1 ++
0855 1 FUNCTIONAL DESCRIPTION:
0856 1
0857 1 The SET MAIL_DIRECTORY command specifies where MAIL should put
0858 1 all its .MAI and .MSG files.
0859 1
0860 1 --
0861 2 BEGIN
0862 2
0863 2 BIND
0864 2 SUB_STRING = DYN_DESC[DSC$A_POINTER] : REF VECTOR[,BYTE];
0865 2
0866 2 LOCAL
0867 2 TMPBUF : VECTOR[NAM$C_MAXRSS,BYTE],
0868 2 PTR,
0869 2 STATUS,
0870 2 LOG_FLAG,
0871 2 DESC : VECTOR[2, LONG],
0872 2 END_CHAR : REF VECTOR[,BYTE];
0873 2
P 0874 2 IF_ERR(CLISGET_VALUE(SD_FILE,COMMAND_DESC);,
0875 2 RETURN TRUE);
0876 2
0877 2 IF .COMMAND_DESC[DSC$W_LENGTH] EQL 0
0878 2 THEN RETURN TRUE;
0879 2
P 0880 2 IF_ERR(CHECK_STRING(COMMAND_DESC,NAM$C_MAXRSS);,
0881 2 RETURN TRUE);
0882 2
0883 2 LOG_FLAG = CLISPRESENT(SD_LOG);
0884 2
0885 2 Syntax check that the parameter is indeed a sub-directory name
0886 2
0887 2 STR$COPY_DX(DYN_DESC,COMMAND_DESC);
0888 2 END_CHAR = SUB_STRING[.COMMAND_DESC[DSC$W_LENGTH]-1];
0889 2 IF NOT CH$FAIL(CH$FIND_CH(.COMMAND_DESC[DSC$W_LENGTH],
0890 2 .COMMAND_DESC[DSC$A_POINTER],%C'-'))
0891 2 OR NOT CH$FAIL(CH$FIND_SUB(.COMMAND_DESC[DSC$W_LENGTH],
0892 2 .COMMAND_DESC[DSC$A_POINTER],2,UPLIT('..'))
0893 2 THEN RETURN SIGNAL(MAIL$ILLSUBDIR,1,COMMAND_DESC);
0894 4 IF ((.SUB_STRING[0] EQL %C'[')
0895 4 AND (.END_CHAR[0] NEQ %C']'))
0896 4 OR ((.SUB_STRING[0] EQL %C'<')
0897 4 AND (.END_CHAR[0] NEQ %C'>'))
0898 2 THEN RETURN SIGNAL(MAIL$NOTSUBDIR,1,COMMAND_DESC);
0899 2
0900 3 IF (.SUB_STRING[0] EQL %C'[')
0901 3 THEN SUB_STRING[0] = %C'<';
0902 3 IF (.END_CHAR[0] EQL %C']')
0903 3 THEN END_CHAR[0] = %C'>';
0904 3 IF (.SUB_STRING[0] NEQ %C'<') OR (.SUB_STRING[1] NEQ %C'.')
0905 3 OR (.END_CHAR[0] NEQ %C'>')
0906 2 THEN RETURN SIGNAL(MAIL$NOTSUBDIR,1,COMMAND_DESC);
0907 2
P 0908 2 IF_ERR(MAIL$GET_USER(MAIL$Q_USERNAME);,
0909 2 RETURN TRUE);
```



```
0910 2
0911 2 DESC[0] = (MAIL$T UAF BUFF[UAF$T_DEFDEV])<CS_COUNT>;
0912 2 DESC[1] = (MAIL$T UAF BUFF[UAF$T_DEFDEV])<CS_STRING>;
0913 2 STR$COPY DX(DYN_DESC_2,DESC);
0914 2 DESC[0] = (MAIL$T UAF BUFF[UAF$T_DEFDIR])<CS_COUNT>;
0915 2 DESC[1] = (MAIL$T UAF BUFF[UAF$T_DEFDIR])<CS_STRING>;
0916 2 STR$APPEND(DYN_DESC_2,DESC);
0917 2
0918 2 PTR = CH$MOVE(.DYN_DESC_2[DSC$W_LENGTH]-1,.DYN_DESC_2[DSC$A_POINTER],TMPBUF);
0919 2 PTR = CH$MOVE(.DYN_DESC[DSC$W_LENGTH]-1,.DYN_DESC[DSC$A_POINTER]+1,.PTR);
0920 2
0921 2 DESC[0] = .PTR - TMPBUF;
0922 2 DESC[1] = TMPBUF;
0923 2 END_CHAR = .DYN_DESC_2[DSC$W_LENGTH] + .DYN_DESC_2[DSC$A_POINTER]-1;
0924 2 IF .END_CHAR[0] EQL '%C%'
0925 2 THEN BEGIN
0926 2     END_CHAR = .DESC[0] + .DESC[1] - 1;
0927 2     END_CHAR[0] = '%C%';
0928 2 END;
0929 2
0930 2 First create the directory. Then, rename MAIL.MAI into that directory.
0931 2 If that fails, give up. If that succeeds, then update VMSMAIL.DAT to
0932 2 contain the subdirectory. After that, rename the rest of the *.mai
0933 2 files to the new directory. By doing it this way, the time that the
0934 2 mail.mai file and vmsmail.dat are out of synch is minimized. Note
0935 2 that errors on the rename of the rest of the files are not dealt with
0936 2 in a meaningful way.
0937 2
0938 2 IF NOT (STATUS = LIB$CREATE DIR(DESC,MAIL$L MYUIC,%REF(-1),%REF(DIR_PRO)))
0939 2 THEN RETURN SIGNAL(MAIL$ DIRNOTCRE,1,DESC,.STATUS)
0940 2 ELSE SIGNAL((IF .STATUS EQL SSS_CREATED
0941 2     THEN MAIL$_CREATED
0942 2     ELSE MAIL$_EXISTS),1,DESC);
0943 2
P 0944 2 IF_ERR(SET_NEW_DIR(COMMAND_DESC,DESC,.LOG_FLAG);,
0945 2     RETURN .STATUS);
0946 2
0947 2 RETURN TRUE
0948 1 END;
```

00 00 2E 2E 0074E .BLKB 2
00750 P.AAZ: .ASCII \..\<0><0>

SUB_STRING= DYN_DESC+4

		OFFC 00000	.ENTRY	MAIL\$CMD SET DIR, Save R2,R3,R4,R5,R6,R7,-	0853
				R8,R9,R10,R11	
5B	F8BA	CF 9E 00002	MOVAB	SD FILE, R11	
5A	00000000	00 9E 00007	MOVAB	COMMAND_DESC, R10	
5E	FEF0	CE 9E 0000E	MOVAB	-272(SPT, SP	
		5A DD 00013	PUSHL	R10	0875
		5B DD 00015	PUSHL	R11	
00000000G	00	02 FB 00017	CALLS	#2, CLISGET_VALUE	
	04	50 E9 0001E	BLBC	STATUS, 1\$	

			00	04	000D7	RET		
			01	9F	000D8	PUSHAB	MAIL\$Q USERNAME	0909
			50	FB	000DE	CALLS	#1, MAIL\$GET_USER	
			00EF	31	000E8	BLBS	STATUS, 14\$	
			00	D0	000EB	BRW	19\$	
			74	9A	000F2	14\$:	MOVL MAIL\$T UAF_BUFF, R0	0911
			75	9E	000F7	MOVZBL	116(R0), DESC	
			08	9F	000FC	MOVAB	117(R0), DESC+4	0912
			10	9F	000FF	PUSHAB	DESC	0913
			02	FB	00102	PUSHAB	DYN_DESC 2	
			00	D0	00109	CALLS	#2, -STR\$COPY DX	
			0094	9A	00110	MOVL	MAIL\$T UAF_BUFF, R0	0914
			0095	9E	00116	MOVZBL	148(R0), DESC	
			08	9F	0011C	MOVAB	149(R0), DESC+4	0915
			10	9F	0011F	PUSHAB	DESC	0916
			02	FB	00122	PUSHAB	DYN_DESC 2	
			10	3C	00129	CALLS	#2, -STR\$APPEND	
			FF	9E	0012D	MOVZWL	DYN_DESC 2, R7	0918
			14	D0	00131	MOVAB	-1(R7), R0	
			50	28	00135	MOVL	DYN_DESC_2+4, R8	
			08	3C	0013A	MOVAB	R0, -(R8), TMPBUF	
			51	D7	0013E	MOVZWL	DYN_DESC, R1	0919
			0C	D0	00140	DECL	R1	
			51	28	00144	MOVL	DYN_DESC+4, R0	
			10	9E	00149	MOVZWL	R1, -1(R0), (PTR)	
			53	C3	0014D	MOVAB	TMPBUF, R0	0921
			10	9E	00152	SUBL3	R0, PTR, DESC	
			FF	9E	00157	MOVAB	TMPBUF, DESC+4	0922
			66	91	0015C	MOVAB	-1(R8)[R7], END_CHAR	0923
			0E	12	00160	CMPB	(END_CHAR), #93	0924
			AE	C1	00162	BNEQ	15\$	
			FF	9E	00168	ADDL3	DESC+4, DESC, R0	0926
			5D	8F	0016C	MOVAB	-1(R0), END_CHAR	
			04	8F	00170	MOVAB	#93, (END_CHAR)	0927
			04	9F	00176	MOVZWL	#65416, 4(TSP)	0938
			01	CE	00179	PUSHAB	4(SP)	
			AE	9F	0017D	MNEGL	#1, 4(SP)	
			00	9F	00180	PUSHAB	4(SP)	
			14	9F	00186	PUSHAB	MAIL\$L_MYUIC	
			04	FB	00189	PUSHAB	DESC	
			50	E8	00190	CALLS	#4, LIB\$CREATE_DIR	
			50	DD	00193	BLBS	STATUS, 16\$	
			0C	9F	00195	PUSHL	STATUS	0939
			01	DD	00198	PUSHAB	DESC	
			8F	DD	0019A	PUSHL	#1	
			04	FB	001A0	PUSHL	#8262338	
			04	04	001A7	CALLS	#4, LIB\$SIGNAL	
			AE	9F	001AB	RET		
			01	DD	001AB	PUSHAB	DESC	0940
			50	D1	001AD	PUSHL	#1	
			08	12	001B4	CPL	STATUS, #1561	
			8F	DD	001B6	BNEQ	17\$	
			06	11	001BC	PUSHL	#8261747	
			8F	DD	001BE	BRB	18\$	
			03	FB	001C4	PUSHL	#8262291	
			59	DD	001CB	CALLS	#3, LIB\$SIGNAL	
						PUSHL	LOG_FLAG	0945

FD41	CF	OC	AE	9F	001CD	PUSHAB	DESC
	03		5A	JD	001D0	PUSHL	R10
	50		03	FB	001D2	CALLS	#3, SET_NEW_DIR
	50		50	E9	001D7	BLBC	STATUS, -20\$
			01	D0	001DA	MOVL	#1, R0
				04	001DD	RET	
					19\$:		
					20\$:		

0947
0948

; Routine Size: 478 bytes, Routine Base: \$CODES + 0754


```
824 0949 1 GLOBAL ROUTINE MAIL$CMD_SET_NODIR =
825 0950 1 ++
826 0951 1 FUNCTIONAL DESCRIPTION:
827 0952 1
828 0953 1 Cancel any SET MAIL_DIRECTORY setting. Cause the mail directory
829 0954 1 to be SYS$LOGIN:
830 0955 1
831 0956 1 --
832 0957 2 BEGIN
833 0958 2
834 0959 2 LOCAL
835 0960 2 DESC : VECTOR[2, LONG],
836 0961 2 LOGIN_DESC : VECTOR[2, LONG],
837 0962 2 LOG_FLAG;
838 0963 2
839 0964 2
840 0965 2 If no mail directory set, then there is nothing to do
841 0966 2
842 0967 2 IF .MAIL$L_USR$SYSBUF[VMD_B_DIRLNG] EQL 0
843 0968 2 THEN RETURN TRUE;
844 0969 2
845 0970 2 LOG_FLAG = CL$PRESENT(SD_LOG);
846 0971 2
847 0972 2
848 0973 2 Get uaf record and get descriptor for logindev:[logindir]
849 0974 2
850 P 0975 2 IF_ERR(MAIL$GET_USER(MAIL$Q_USERNAME));,
851 0976 2 RETURN TRUE;
852 0977 2
853 0978 2 DESC[0] = (MAIL$T_UAF_BUFF[UAF$T_DEFDEV])<CS_COUNT>;
854 0979 2 DESC[1] = (MAIL$T_UAF_BUFF[UAF$T_DEFDEV])<CS_STRING>;
855 0980 2 STR$COPY_DX(COMMAND_DESC, DESC);
856 0981 2 DESC[0] = (MAIL$T_UAF_BUFF[UAF$T_DEFDIR])<CS_COUNT>;
857 0982 2 DESC[1] = (MAIL$T_UAF_BUFF[UAF$T_DEFDIR])<CS_STRING>;
858 0983 2 STR$APPEND(COMMAND_DESC, DESC);
859 0984 2 DESC[0] = 0;
860 0985 2 DESC[1] = DESC[0];
861 0986 2
862 0987 2
863 0988 2 Rename all the mail files to the login directory.
864 0989 2 Note that if the login directory doesn't exist, this command won't create it.
865 0990 2
866 0991 2 SET_NEW_DIR(DESC, COMMAND_DESC, .LOG_FLAG);
867 0992 2
868 0993 2 RETURN TRUE
869 0994 1 END;
```

```
003C 00000
55 00000000G 00 9E 00002
54 F6E1 CF 9E 00009
53 00000000' 00 9E 0000E
5E 10 C2 00015
50 00000000G 00 D0 00018
```

```
.ENTRY MAIL$CMD SET NODIR, Save R2,R3,R4,R5
MOVAB MAIL$T_UAF_BUFF, R5
MOVAB SD_LOG, R4
MOVAB COMMAND_DESC, R3
SUBL2 #16, SP
MOVL MAIL$L_USR$SYSBUF, R0
```

```
: 0949
:
:
:
: 0967
```

MAIL\$SET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

L 11
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]SET.B32;1

Page 40
(21)

		41	A0	95	0001F	TSTB	65(R0)	
			64	13	00022	BEQL	1\$	
			54	DD	00024	PUSHL	R4	0970
00000000G	00		01	FB	00026	CALLS	#1, CL\$PRESENT	
	52		50	D0	0002D	MOVL	R0, LOG_FLAG	
		00000000G	00	9F	00030	PUSHAB	MAIL\$Q_USERNAME	0976
00000000G	00		01	FB	00036	CALLS	#1, MAIL\$GET_USER	
	48		50	E9	0003D	BLBC	STATUS, 1\$	
	50		65	D0	00040	MOVL	MAIL\$T_UAF_BUFF, R0	0978
08	AE	74	A0	9A	00043	MOVZBL	116(R0), DESC	
0C	AE	75	A0	9E	00048	MOVAB	117(R0), DESC+4	0979
		08	AE	9F	0004D	PUSHAB	DESC	0980
			53	DD	00050	PUSHL	R3	
00000000G	00		02	FB	00052	CALLS	#2, STR\$COPY_DX	
	50		65	D0	00059	MOVL	MAIL\$T_UAF_BUFF, R0	0981
08	AE	0094	C0	9A	0005C	MOVZBL	148(R0), DESC	
0C	AE	0095	C0	9E	00062	MOVAB	149(R0), DESC+4	0982
		08	AE	9F	00068	PUSHAB	DESC	0983
			53	DD	0006B	PUSHL	R3	
00000000G	00		02	FB	0006D	CALLS	#2, STR\$APPEND	
		08	AE	D4	00074	CLRL	DESC	0984
0C	AE	08	AE	9E	00077	MOVAB	DESC, DESC+4	0985
			52	DD	0007C	PUSHL	LOG_FLAG	0991
			53	DD	0007E	PUSHL	R3	
		10	AE	9F	00080	PUSHAB	DESC	
FCB2	CF		03	FB	00083	CALLS	#3, SET_NEW_DIR	
	50		01	D0	00088	MOVL	#1, R0	0993
			04	0008B	1\$: RET			0994

; Routine Size: 140 bytes, Routine Base: \$CODE\$ + 0932

MAIL\$SET
V04-000

VAX/VMS MAIL UTILITY SET COMMAND

M 11
16-Sep-1984 01:20:06
14-Sep-1984 12:42:32

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[MAIL.SRC]SET.B32;1

Page 41
(22)

: 871 0995 1 END
: 872 0996 0 ELUDOM

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	24	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	2494	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	46	0	1000	00:01.4
\$255\$DUA28:[MAIL.OBJ]MAILDEF.L32;1	457	23	5	26	00:00.4

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS\$;SET/OBJ=OBJ\$;SET MSRC\$;SET/UPDATE=(ENH\$;SET)

: Size: 2312 code + 206 data bytes
: Run Time: 00:26.1
: Elapsed Time: 01:36.7
: Lines/CPU Min: 2288
: Lexemes/CPU-Min: 27842
: Memory Used: 200 pages
: Compilation Complete

0231 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

READ
LIS

SENDMSG
LIS

PRINT
LIS

SET
LIS

SPAWNCD
LIS

MANAGE

ALFMINT
LIS

SYSMAT
LIS

SHOW
LIS